ByteDance | Seed

Stable Autoregressive Speech Generation with Low-Frame-Rate High-Dimensional Continuous Tokens

Yi Luo, **Rongzhi Gu**, **Jixun Yao***

ByteDance Seed

*Work done during an internship at ByteDance Seed.

Abstract

Balancing sequence length, representational capacity, and long-horizon stability is a central problem in autoregressive (AR) speech and audio generation. Representations with higher frame rates or greater capacity can preserve more signal detail, but they also make streaming generation more vulnerable to distribution drift and AR error accumulation. Conversely, shorter and more compressed representations simplify AR modeling, but their limited bandwidth may discard important components and constrain the upper bound of reconstruction fidelity and generation quality. We ask whether a low-frame-rate, high-dimensional, high-bandwidth continuous representation can be co-designed with a streaming generation framework to support robust high-fidelity reconstruction, strong single-token predictability, and superior long-horizon stability. We decompose this goal into two coupled problems: what geometric and statistical properties a high-dimensional representation space should have, and how an AR continuous-token generator should be structured to resist error accumulation. Accordingly, we propose Locodec, a locally encoded codec that shapes its representation space to improve the interpolatability of a lower-dimensional core manifold and the identifiability of the native high-dimensional coordinates, thereby improving the predictability of high-dimensional high-bandwidth tokens. We also propose MP-ELD, a single-token AR flow-matching framework that uses multi-path information routing and residual classifier-free guidance to mitigate error accumulation. Experiments with 8-Hz, 768-dimensional tokens show that our design preserves reconstruction quality, improves single-token predictability, achieves competitive WER, and maintains stable long-form synthesis, without using external SSL/ASR models, pretrained text language models, or post-training stages.

Date: August 3, 2026
Correspondence: Yi Luo at yl3364@columbia.edu

1 Introduction

Audio is one of the most natural interfaces for human–machine interaction, and the ability to perform stable streaming understanding and generation has become increasingly important for modern audio and multimodal foundation models [51, 98, 109]. From the input side of such systems, a representation that preserves more information generally provides a higher ceiling for downstream understanding. If the representation has already discarded task-relevant cues through aggressive lossy compression, no subsequent model can recover them reliably. In this sense, rich or near-complete representations allow the model to learn which factors are useful for a given task, rather than forcing it to operate on a representation that may not contain those factors

in the first place. From the output side, however, the same preference for high-information representations creates a different difficulty. A higher-bandwidth or higher-dimensional target space is usually harder to predict, and prediction errors in an autoregressive (AR) system are fed back as future context. Small local prediction errors may therefore accumulate over time, leading to drift in loudness, timbre, speaking rate, or spectral quality, and in severe cases to unstable or collapsed audio outputs. This creates a basic tension: representations that are desirable for general understanding and high-fidelity reconstruction can be difficult to use as stable generation targets. Balancing representation capacity, AR stability, and computational efficiency is therefore a central design problem in streaming audio systems.

As text language models and audio–text datasets continue to improve, transcription, captioning, and other understanding-oriented capabilities have made steady progress in large audio models [28, 34, 36, 42, 108, 112, 114, 122]. By contrast, stable and efficient high-fidelity streaming generation remains a bottleneck: it directly determines the perceptual quality of human-facing interaction, and it also affects subsequent machine-side understanding in closed-loop systems. The problem becomes more pronounced as interactive scenarios move from second-scale responses to minute- or hour-scale sessions, while practical context length and computation budgets remain finite. Ideally, one would like to use a representation that is informative enough to support high-fidelity reconstruction, short enough to reduce the AR horizon, and simple enough to model without expensive local predictors or post-processing modules. These requirements, however, seem to form an unfavorable triangle: high information content, low AR error accumulation, and low model complexity are difficult to satisfy simultaneously.

This paper explores whether this triangle can be relaxed by jointly designing the representation space and the AR generative model. Motivated by recent progress in raw signal-space prediction [63, 103], representation-space analysis [117, 124], and AR flow-matching systems [53, 64], we study a low-frame-rate, high-dimensional continuous-token formulation together with its corresponding AR generation framework. Instead of relying on externally pretrained self-supervised learning (SSL) or automatic speech recognition (ASR) models to define a "semantic" space as a supervision signal for shaping the representation space, we train a reconstruction-first tokenizer whose latent geometry is shaped directly. The resulting tokenizer, Locodec, produces spherical continuous tokens whose high-dimensional space is organized around a lower-dimensional interpolatable core manifold, while its native coordinates are encouraged to develop an energy hierarchy that improves per-token identifiability. This design aims to keep the bandwidth and reconstruction ceiling of a high-capacity representation, while making single-token prediction substantially easier for the generative model.

To address the remaining long-horizon instability of high-information AR generation, we further examine how error accumulation arises during guided streaming synthesis. Our empirical observations suggest that a major source of accumulated drift is instability under classifier-free guidance (CFG), especially when partially overlapping acoustic cues are carried by different guidance paths and are amplified inconsistently. Rather than suppressing guidance globally, we aim to reduce such conflicts by encouraging different types of information to be routed through functionally distinct pathways. We therefore propose MP-ELD, a multi-path encoder–LM–decoder framework for AR flow matching. Through explicit information routing and training-time path dropout, MP-ELD encourages different aspects of the audio state to be represented by different conditioning pathways. This allows guidance to be applied as structured residual corrections, so that different factors, such as acoustic consistency and external-condition alignment, can be strengthened separately. In practice, even without pretrained text language models providing a strong semantic inductive bias, this design allows a generative model trained from scratch to form distinguishable acoustic-state and alignment-related pathways. This separation makes it possible to control their CFG scales independently, thereby mitigating accumulated acoustic drift and improving stability in long-form synthesis.

The rest of this paper is organized as follows. Section 2 discusses the main assumptions and design principles behind our model design, including the roles of bitrate, representation-space geometry, and AR stability. Section 3 presents the methodology used to shape the tokenizer latent space and to construct the flow-matching bridge and training objective. Section 4 describes the concrete model architectures of the tokenizer and the AR generative model. Section 5 reports the experimental results on tokenizer reconstruction quality and generative model performance on the Seed-TTS-eval dataset. Section 6 concludes the paper.

2 Main Assumptions and Design Principles

In this section, we revisit several existing design choices for tokenizers and generative models, discuss their advantages and limitations, and then motivate the principles that guide our overall design.

2.1 Frame rate, bitrate, and reconstruction fidelity

Due to the nature of audio signals, audio processing has long been confronted with the challenge of modeling extremely long sequences [12, 35, 40, 46, 50, 70, 73, 75, 81, 83, 88, 93, 100, 119]. For a fixed audio duration, the token sequence length is determined by the token frame rate, which directly controls the temporal span and granularity covered by each token. Together with the information capacity of each token, it also determines the amount of information that can be allocated per unit time, i.e., the bandwidth. While tokenizer and generative model designs have been extensively explored for moderate bitrate configurations [21, 33, 34, 52, 58, 66, 102, 123], a major line of research in recent years has focused on achieving the highest possible reconstruction fidelity under aggressively reduced frame rate and bitrate settings [37, 44, 54, 55, 84, 105]. Lowering the frame rate is one of the most direct ways to reduce the complexity of the downstream sequence model, while lowering the bitrate can further simplify the generative modeling problem and reduce computational cost. For example, replacing residual vector quantization (RVQ) [121] with single-stage vector quantization (VQ) [101] removes the need to model a residual hierarchy, thereby reducing both the required computation and the effective parameter budget.

However, from the perspective of lossless compression, low bitrate and high reconstruction fidelity are fundamentally difficult, and often impossible, to achieve simultaneously. As a result, existing tokenizers typically preserve only a subset of the most important signal attributes, such as content, timbre, or pitch, while (intentionally or unintentionally) discarding other factors that are more difficult to retain faithfully. Under such a training objective, many low-bitrate tokenizers are often closer to a resynthesis system than to a strict codec. Consequently, in certain regimes or tasks, such as high-fidelity generation or local editing, the upper bound of performance may be directly constrained by the tokenizer bitrate and by the corresponding training paradigm.

Conversely, when one increases bitrate in order to improve the information capacity of discrete tokens, for instance via deeper RVQ hierarchies or larger finite scalar quantization (FSQ) bitrates [77], the resulting latent space gradually becomes closer to a continuous one. From the modeling perspective, when a discrete tokenization scheme such as RVQ uses many residual levels, and the generative model must unroll these levels hierarchically during prediction, e.g., via delay-pattern modeling [29] or residual-quantization transformers (RQTransformer) [59], the model complexity increases substantially. Recent work has therefore increasingly explored continuous representations as a means of enabling higher-bandwidth modeling [25, 39, 68, 76, 86, 111]. On the one hand, under the same model architecture, frame rate, and token dimensionality, one can often significantly improve reconstruction fidelity simply by removing the quantization step altogether. On the other hand, from the perspective of iterative prediction, the residual hierarchy in high-bitrate discrete tokenization is, to some extent, analogous to the iterative denoising or function evaluation process in flow-based continuous generation. In this sense, once bitrate is increased sufficiently, continuous tokenization becomes a natural modeling direction.

At the same time, both high-bitrate discrete tokens and continuous tokens have substantially more target-space degrees of freedom than low-bitrate tokens, which in practice can make AR systems more vulnerable to accumulated prediction errors and exposure bias [90]. Moreover, when the tokenizer frame rate is high and the AR sequence is correspondingly long, this accumulation can be further amplified. Therefore, for high-bandwidth tokenization, reducing the frame rate, or more generally reducing the effective frame rate seen by the AR model, becomes a necessary means of controlling AR error accumulation.

2.2 Single high-dimensional space versus product-structured space

When discussing low-frame-rate, high-bandwidth continuous tokens, a natural design question arises: should one directly construct tokens that are natively low in frame rate but higher in dimensionality, so that the AR model operates directly at the token rate; or should one instead construct tokens that are higher in frame rate

but lower in dimensionality, and then group nearby tokens so that the AR model effectively operates at the grouped frame rate?

Suppose that the overall tokenizer compression ratio is fixed across these two choices. Then, from the perspective of the AR model, the main difference is whether the model sees a single high-dimensional space or a product-structured space composed of multiple lower-dimensional spaces. In most existing approaches, the latter is more common: each local group of low-dimensional tokens is treated as a short local sequence, which is first compressed into a single embedding before entering the language model, and is then decoded back into the next local token sequence at prediction time [53, 54, 126, 127]. This strategy is related to what prior work on efficient speech enhancement and separation referred to as a context codec, namely a local compression/decompression mechanism designed to shorten the effective input length of the sequence model [71]. By contrast, among systems that attempt to operate at a natively low frame rate, one rarely observes continuous token configurations that are simultaneously very high-dimensional (e.g., 256 dimensions or above) and sufficiently high in bandwidth to support high reconstruction fidelity. One possible reason is the geometric and statistical difficulty of modeling such a high-dimensional space, together with the well-known "curse of dimensionality" [3]. As a consequence, a grouped low-dimensional product-structured representation appears, at least superficially, to be a more practical way of constructing high-capacity tokens.

Nevertheless, this observation immediately raises another question. If one must already introduce an additional encoder to compress a group of low-dimensional tokens into a single representation for the language model, and then introduce a corresponding decoder to reconstruct the next token group from the language-model output, why should this compression–decompression process be placed inside the generative model rather than inside the tokenizer itself? In other words, if the generative model must reshape a high-frame-rate, low-dimensional token sequence into a low-frame-rate, high-dimensional latent space, and then invert that reshaping during prediction, does this not suggest that one may instead train a tokenizer that directly produces such a native low-frame-rate, high-dimensional token space? Ideally, such a tokenizer would preserve sufficient bandwidth to maintain strong reconstruction fidelity, while eliminating the need for additional local encoding and decoding modules inside the generative model, especially prediction-time modules such as local DiTs that substantially increase computational cost and architectural complexity.

From the viewpoint of the generative model input, compressing a group of low-dimensional tokens can be implemented either inside the tokenizer or inside the generative model, and these two choices are not fundamentally different. However, from the viewpoint of the generative model output, the difference is substantial. A local token group, being itself a short sequence, can be processed by a local sequence model that has sufficient capacity to progressively lift and reshape the local representation, thereby making the corresponding product-structured representation considerably easier to model. In contrast, a single high-dimensional token is not itself a sequence and therefore cannot benefit from such sequential lifting; its representational expansion inside the model is much more limited, since excessively large hidden dimensions would directly increase model size, training difficulty, and computational cost. Therefore, the key challenge is not simply to use a high-dimensional space, but to construct such a space so that the generative model can perform stable and efficient single high-dimensional token prediction, ideally reaching performance comparable to, or better than, sequential prediction over grouped low-dimensional tokens, at significantly lower computational cost.

2.3 Disentangled representation space

One of the most common strategies in prior work for reducing generative modeling difficulty, applicable to both discrete and continuous tokens, is to explicitly construct a coarse-to-fine representation space, most often in the form of semantic–acoustic disentanglement [10, 14, 24, 37, 67, 80, 104, 115]. In such representations, the semantic component typically serves as a compact core representation that preserves the main attributes of the signal, and may come from the hidden space of a pretrained SSL model [9, 20, 27, 49] or an ASR model [18, 87]. The acoustic component then complements this semantic representation by providing additional detail, often through a residual or hierarchical structure. By feeding the language model with a more clustered, more constrained, or otherwise lower-freedom semantic representation, AR error accumulation can be effectively reduced on the input side; and on the output side, the semantic embedding itself can simplify and stabilize the prediction. Similar trends have also appeared increasingly in image generation, where large-scale pretrained

SSL embeddings have been used either as alignment supervision during tokenizer training or directly as tokens themselves, improving the generative performance of continuous, especially high-dimensional continuous, representations [62, 99, 124].

However, relying on such predefined or externally trained semantic embeddings is not a free lunch. On the one hand, semantic embeddings derived from models trained for specific domains or attributes, such as ASR models, inevitably inherit strong biases toward the corresponding target attributes, for example, word or phoneme structure, or even language-dependent biases. Such biases may reduce the capacity of the tokenizer to encode other types of information, such as non-semantic or non-vocal content, forcing those properties to be represented only in the acoustic component. If the generative model then relies heavily on the semantic component in order to reduce AR error accumulation, other types of information may be underrepresented in its input, limiting the model's understanding and generation capabilities. On the other hand, semantic embeddings derived from SSL models are heavily shaped by the SSL training data, model scale, and pretraining objective. Different generative tasks, e.g., speech synthesis versus music generation, may prefer different representation-space properties. If one wishes to build a single task-general representation space, the cost of training a sufficiently large and sufficiently universal SSL model may even exceed that of training the generative model itself.

Therefore, from the perspectives of simplicity and generality, we prefer not to rely on explicitly disentangled semantic–acoustic representations, nor on external large-scale SSL training to shape or constrain the high-dimensional token space. Instead, we aim to let the tokenizer optimize primarily for high-fidelity reconstruction, while making its decoder sufficiently robust to cover the types of prediction errors produced by the generative model, and to use lower-cost mechanisms to induce a high-dimensional space that is itself easier for the generative model to predict.

2.4 Raw input space modeling and the manifold hypothesis

Another line of work that has recently attracted increasing attention is to perform generative modeling directly in the high-dimensional raw signal space, without any separately trained tokenizer. Such approaches are often motivated by the manifold hypothesis, namely the assumption that many real-world high-dimensional signals in fact lie near a lower-dimensional latent manifold [16, 43]. Under this view, directly modeling such signals may be easier than modeling an arbitrary high-dimensional vector that does not inherit such structure, such as a freely learned high-dimensional token. This line of work has shown strong potential in image generation [48, 63, 118], and has also begun to appear in audio generation [23, 41, 125].

However, unlike direct modeling in image pixel space, direct modeling in waveform space might not necessarily be a natural choice for audio. In waveform space, an isotropic Gaussian corruption process, when viewed under an orthonormal time–frequency transform, corresponds to broadly full-band white noise in the frequency domain. Real-world audio signals, by contrast, tend to exhibit substantially lower natural energy in the mid- and high-frequency bands. As a consequence, if the generative model fails to remove the injected noise completely, the residual error in these bands may become perceptually salient under human auditory perception [79, 128]. This may differ from the image case where human sensitivity and tolerance to such noises might have a different mechanism [11, 15, 22], and where the masking effects of additive noise might also be qualitatively different from those in audio [106]. Therefore, the same type of prediction error may have significantly different perceptual consequences in audio and image generation. In addition, although pixel-space modeling in image generation has been empirically shown to achieve performance comparable to, and in some cases better than, latent-space modeling, it may still exhibit artifacts or noise due to the lack of a noise-robust tokenizer decoder that can refine or compensate for prediction errors. Therefore, what we seek is a high-dimensional space that still benefits from the manifold hypothesis, but at the same time possesses a meaningful degree of reconstruction robustness, so that the generative model can learn it more easily due to the presence of a structured and interpolatable low-dimensional manifold, while also retaining enough robustness to prevent generative prediction errors from directly becoming perceptually severe distortions.

2.5 Representational interpolatability and identifiability

Another attribute that affects whether a representation space can be effectively modeled by a generative model is its interpolatability [6, 107]. An interpolatable representation space is often associated with a smoother manifold and better clustering properties, and prior work has observed that such spaces are more likely to lead to better generation performance [7, 30, 32, 96, 113, 120]. However, directly constructing sufficiently strong interpolatability over an entire high-dimensional space is mathematically highly impractical. The capacity of a high-dimensional space grows so rapidly that neither finite training data nor local noise augmentation can densely cover it in any meaningful sense.

We illustrate this point using a simplified spherical-cap covering problem. We consider a high-dimensional sphere and spherical tokens lying on it. We consider the noise robustness of a tokenizer by assuming a "reconstruction-stable basin", where for each token there exists a spherical cap that allows the tokenizer decoder to reconstruct a near-identical raw signal from any token within the cap. We thus ask: how many non-overlapping basins can the sphere $\mathbb{S}^{N-1}$ contain? In an interpolatable space, one would expect reconstruction-stable basins to be sufficiently connected or overlapping at a moderate basin density, so that moving between nearby valid regions does not frequently pass through invalid regions on the sphere. On the other hand, if the number of non-overlapping basins is enormously large and far exceeds the number of training samples, it means that the space has enough capacity for training samples to occupy mutually isolated basins without forcing substantial basin overlap. As a result, the valid regions on the sphere only cover a minor portion of the total capacity, and global interpolatability in this case is nearly impossible.

We formulate this problem via mathematical arguments. Consider the unit sphere

$$\mathbb{S}^{N-1} = \{x \in \mathbb{R}^N : \|x\|_2 = 1\}.$$

For a reference point $e_1 = (1, 0, \ldots, 0)$, define the spherical cap of geodesic half-angle $\theta \in (0, \pi/2)$ as

$$\mathcal{C}_\theta = \left\{x \in \mathbb{S}^{N-1} : \arccos(\langle x, e_1\rangle) \le \theta\right\}.$$

Let

$$\mu_N(\theta)$$

denote the normalized surface measure of this cap. Equivalently, if x is uniformly distributed on $\mathbb{S}^{N-1}$, then

$$\mu_N(\theta) = \Pr\left(\langle x, e_1\rangle \ge \cos\theta\right).$$

The exact expression is

$$\mu_N(\theta) = \frac{\int_0^\theta \sin^{N-2}\phi\, d\phi}{\int_0^\pi \sin^{N-2}\phi\, d\phi}. \tag{1}$$

Equivalently, the first coordinate of a uniformly sampled point on the sphere has density

$$p_N(u) = \frac{\Gamma(N/2)}{\sqrt{\pi}\Gamma((N-1)/2)}(1-u^2)^{(N-3)/2}, \qquad u \in [-1, 1],$$

and therefore

$$\mu_N(\theta) = \int_{\cos\theta}^1 \frac{\Gamma(N/2)}{\sqrt{\pi}\Gamma((N-1)/2)}(1-u^2)^{(N-3)/2}\, du. \tag{2}$$

Applying a standard endpoint Laplace approximation, for any fixed $\theta \in (0, \pi/2)$ we obtain

$$\mu_N(\theta) \sim \frac{(\sin\theta)^{N-1}}{\cos\theta\sqrt{2\pi N}}, \qquad N \to \infty. \tag{3}$$

Thus the reciprocal cap measure satisfies

$$K_{\text{area}}(N, \theta) := \frac{1}{\mu_N(\theta)} \sim \cos\theta\sqrt{2\pi N}\,(\sin\theta)^{-(N-1)}. \tag{4}$$

Table 1 Area scale $K_{\text{area}}(N,\theta) = 1/\mu_N(\theta)$ for spherical caps on $\mathbb{S}^{N-1}$. This quantity is a lower bound on the number of θ-caps required to cover the sphere, and also indicates the exponential area scale associated with angular regions. Values are approximate.

N	$\theta = 30^\circ$	$\theta = 45^\circ$	$\theta = 60^\circ$
4	$\approx 3.5 \times 10^1$	$\approx 1.1 \times 10^1$	≈ 5.1
8	$\approx 7.9 \times 10^2$	$\approx 6.0 \times 10^1$	$\approx 1.2 \times 10^1$
16	$\approx 2.5 \times 10^5$	$\approx 1.3 \times 10^3$	$\approx 4.9 \times 10^1$
32	$\approx 2.6 \times 10^{10}$	$\approx 4.6 \times 10^5$	$\approx 6.1 \times 10^2$
64	$\approx 1.6 \times 10^{20}$	$\approx 4.3 \times 10^{10}$	$\approx 8.6 \times 10^4$
128	$\approx 4.2 \times 10^{39}$	$\approx 2.6 \times 10^{20}$	$\approx 1.2 \times 10^9$
256	$\approx 2.0 \times 10^{78}$	$\approx 6.8 \times 10^{39}$	$\approx 1.7 \times 10^{17}$
512	$\approx 3.3 \times 10^{155}$	$\approx 3.3 \times 10^{78}$	$\approx 2.4 \times 10^{33}$

The essential point is that for every fixed $\theta < 90^\circ$, this quantity grows exponentially in N. Let $\mathcal{N}_{\text{cov}}(N,\theta)$ denote the minimum number of spherical caps of radius θ required to cover $\mathbb{S}^{N-1}$. Since each cap occupies surface fraction $\mu_N(\theta)$, one necessarily has

$$\mathcal{N}_{\text{cov}}(N,\theta) \geq \frac{1}{\mu_N(\theta)} = K_{\text{area}}(N,\theta). \tag{5}$$

Therefore, if one wants every point on the high-dimensional sphere to lie within angle θ of some reconstruction-stable region, the required number of such regions is already at least exponential in N. For non-overlapping basins, let $\mathcal{N}_{\text{pack}}(N,\theta)$ denote the maximum number of disjoint spherical caps of radius θ. The simple volume argument gives

$$\mathcal{N}_{\text{pack}}(N,\theta) \leq \frac{1}{\mu_N(\theta)}.$$

Conversely, a standard maximal-packing argument gives an area-scale lower bound. For $\theta < \pi/4$, choose a maximal collection of disjoint θ-caps. By maximality, the caps with doubled radius 2θ must cover the sphere. If the number of selected caps is M, then

$$M\,\mu_N(2\theta) \geq 1.$$

Since $\mathcal{N}_{\text{pack}}(N,\theta)$ is the maximum number of disjoint θ-caps, we obtain

$$\mathcal{N}_{\text{pack}}(N,\theta) \geq \frac{1}{\mu_N(2\theta)}. \tag{6}$$

Thus, even conservative packing lower bounds can be exponential in dimension. For example, the scale $K_{\text{area}}(N, 60^\circ)$ gives a lower-bound scale for the number of disjoint 30° basins obtainable by such a maximal-packing argument.

To make the scale concrete, Table 1 reports $K_{\text{area}}(N,\theta) = 1/\mu_N(\theta)$ for representative dimensions and angles. The values should be interpreted as area/coverage scales rather than exact packing numbers. They are computed from the exact cap-measure expression when numerically convenient and from the asymptotic expression in Eq. (4) in the large-N regime. Several observations follow. First, even for a relatively large angular tolerance such as 60°, the area scale reaches roughly 10^5 at $N = 64$ and 10^{17} at $N = 256$. For smaller angular tolerances, the numbers become enormous much earlier. Second, these estimates are already sufficient to show that direct dense interpolatability in a high-dimensional spherical token space is not a practical goal. Local noise injection around training samples can improve decoder robustness near the data manifold, but it cannot make the entire high-dimensional sphere densely connected in any global sense. Therefore, if one wishes to induce interpolatability, it is far more plausible to construct it on a lower-dimensional manifold embedded within the high-dimensional space, and then use the geometry of this lower-dimensional manifold to shape the high-dimensional token space.

At the same time, the dimension of the lower-dimensional manifold should not be too small, as interpolatability is only a valid objective when the low-dimensional space itself contains sufficient information for coarse

reconstruction. A representation space can be highly interpolatable simply because it has collapsed most information: if the decoder cannot reconstruct the signal meaningfully, then moving smoothly in that space is of little value. More formally, let X denote the data signal and let an encoder–decoder pair with a latent representation of token dimension d, denoted by $\mathcal{M}_d$, induce reconstructions

$$\widehat{X} = D(E(X)), \qquad E(X) \in \mathcal{M}_d.$$

For a reconstruction loss ℓ, define the best achievable distortion at dimension d as

$$\mathcal{D}^\star(d) = \inf_{E,D} \mathbb{E}\big[\ell(X, D(E(X)))\big], \tag{7}$$

under the architectural, smoothness, regularization, and bandwidth constraints of interest. A meaningful low-dimensional manifold must satisfy

$$\mathcal{D}^\star(d) \le \varepsilon_{\text{rec}}$$

for a reconstruction tolerance ε_{rec} relevant to the target task. If d is too small, this condition fails regardless of how smooth or interpolatable the latent space appears.

The same issue can be expressed in terms of over-clustering. Suppose that the latent space is approximately spherical with token dimension d. At an angular resolution ρ, the number of disjoint stable regions is upper-bounded by the cap-area scale

$$K_{\text{area}}(d, \rho) = \frac{1}{\mu_d(\rho)}.$$

Let $M_{\text{data}}(\varepsilon)$ denote the effective number of perceptually distinguishable signal states at tolerance ε. If

$$K_{\text{area}}(d, \rho) \ll M_{\text{data}}(\varepsilon),$$

then even this optimistic upper bound is insufficient to assign distinct stable regions to all perceptually distinguishable signal states, and many distinct signals must therefore be mapped into the same or nearby latent regions. Under a smooth or noise-robust decoder, nearby latent points tend to decode to nearby reconstructions, and excessive clustering may further lead to averaged or over-smoothed outputs. Consequently, an excessively low-dimensional manifold may force unrelated or only weakly related signals to be clustered together, leading to over-smoothing, loss of diversity, or mode merging. Moreover, if the high-dimensional token space is explicitly aligned with this low-dimensional manifold, such over-clustering can also bias the information layout of the high-dimensional space and make the generative model inherit the same smoothing or diversity-loss tendency. In practice, this suggests choosing a moderate dimension: large enough to support meaningful reconstruction and sufficient latent capacity, but small enough that the induced geometry remains substantially more interpolatable than the original high-dimensional token sphere.

Besides interpolatability, identifiability is another factor that may affect modeling difficulty. In many audio processing systems, time–frequency representations play a central role partly because they introduce an energy-based inductive bias: perceptually important or structurally dominant components often occupy higher-energy regions, while finer details occupy lower-energy regions. Under an additive isotropic corruption model

$$y_i = x_i + \epsilon_i, \qquad \epsilon_i \sim \mathcal{N}(0, \sigma^2),$$

a feature coordinate with energy $e_i = \mathbb{E}[x_i^2]$ has an effective coordinate-wise signal-to-noise ratio that scales as

$$\text{SNR}_i \propto \frac{e_i}{\sigma^2}.$$

Higher-energy components therefore remain identifiable under stronger corruption. Motivated by this observation, we define identifiability in the present context as an inductive bias that highlights core content through energy allocation in feature space. If a high-dimensional token space can be shaped so that more reconstruction-critical information tends to occupy higher-energy components, then the generative model may more easily infer and preserve the core content of each token even under prediction noise.

In summary, interpolatability and identifiability impose complementary requirements. Interpolatability suggests that the high-dimensional token space should be constrained by a lower-dimensional manifold, while

identifiability suggests that the high-dimensional coordinates should develop an energy hierarchy that makes important information more robust and easier to recover. Together, these two attributes may make the high-dimensional space easier for a generative model to learn and predict.

2.6 Core objectives of tokenizer design

The discussion above suggests a set of core objectives for the type of low-frame-rate, high-dimensional token space that we would like to construct:

- it should be constrained by, or organized around, a lower-dimensional manifold embedded within the high-dimensional space, where this manifold is interpolatable and well-clustered while still having sufficient dimensionality to support meaningful reconstruction and avoid excessive over-clustering or smoothing;
- it should have sufficiently strong reconstruction robustness under noise;
- it should not rely on semantic or SSL spaces defined by external models;
- it should have sufficiently strong per-token identifiability.

3 Methodology

In this section, we describe the concrete methods used to design our tokenizer and generative model, following the assumptions and design principles discussed above.

3.1 Concentration of statistics in high-dimensional spaces

A standard variational autoencoder regularizes its approximate posterior toward a standard Gaussian prior through the KL term. Let

$$g \sim \mathcal{N}(0, I_N), \qquad g \in \mathbb{R}^N.$$

Then its squared norm satisfies

$$\|g\|_2^2 = \sum_{i=1}^{N} g_i^2 \sim \chi_N^2,$$

with

$$\mathbb{E}\big[\|g\|_2^2\big] = N, \qquad \mathrm{Var}\big(\|g\|_2^2\big) = 2N.$$

As a consequence,

$$\frac{\|g\|_2}{\sqrt{N}} \to 1 \quad \text{in probability as } N \to \infty,$$

which implies that high-dimensional Gaussian vectors concentrate near the sphere of radius $\sqrt{N}$. Therefore, in sufficiently high dimensions, a standard Gaussian latent can be viewed, to a good approximation, as living on a thin spherical shell. This naturally motivates learning a spherical token space, i.e., a token space with approximately fixed norm. From the perspective of prediction, for any target token $x \in \mathbb{R}^N$ and prediction $\hat{x} \in \mathbb{R}^N$, the prediction error can be decomposed in polar form into radial and angular components. If both the target and the prediction are constrained or normalized to the same sphere, the radial degree of freedom is removed, and the remaining prediction error is determined by angular discrepancy. This is attractive for two reasons. First, the tokenizer decoder only needs to learn robustness with respect to angular perturbations. Second, the generative model can in principle remove one source of exposure bias by predicting within a fixed-norm space, rather than having to model both the norm and the direction of the token.

For these reasons, in the remainder of this paper we assume a high-dimensional spherical token space as the default setting.

3.2 Shaping decoder noise robustness

Prior work on spherical VAEs often adopts the von Mises–Fisher (vMF) distribution as the prior over spherical latent variables [31]. However, in practice, the KL term and reparameterization for the vMF distribution are typically more cumbersome to implement, and often require additional approximations or specialized estimators [56]. Here we instead consider a simpler alternative: rather than imposing an explicit prior through a KL term, we shape the information bottleneck and decoder robustness by injecting stronger noise directly into the token space without constraining the token prior distribution.

For a spherical token space, the most natural corruption mechanism is a random rotation on the sphere. Let

$$x \in \mathbb{S}^{N-1}(R) = \{u \in \mathbb{R}^N : \|u\|_2 = R\}$$

be a clean token. To construct a noisy token on the same sphere, we first sample a random tangent direction. Concretely, let

$$\xi \sim \mathcal{N}(0, I_N),$$

and project it onto the tangent space of the sphere at x:

$$u = \xi - \frac{\langle \xi, x \rangle}{\|x\|_2^2} x. \tag{8}$$

Then

$$u \in T_x\mathbb{S}^{N-1}(R), \qquad \langle u, x \rangle = 0.$$

We obtain a unit tangent direction at x by normalizing

$$\bar{u} = \frac{u}{\|u\|_2}.$$

We then sample a rotation angle $\theta \in [0, \pi/2]$ and define the rotated noisy token through the spherical exponential map:

$$x^{\text{rot}} = \cos\theta\, x + \sin\theta\, R\bar{u}. \tag{9}$$

Since x and $\bar{u}$ are orthogonal, it follows directly that

$$\|x^{\text{rot}}\|_2^2 = \cos^2\theta\, \|x\|_2^2 + \sin^2\theta\, R^2 = R^2,$$

so x^{rot} remains exactly on the same sphere. Moreover, the geodesic angle between x and x^{rot} is precisely θ.

In practice, we sample a scalar

$$s \sim \text{Beta}(1, 2)$$

and set

$$\theta = \frac{\pi}{2} s.$$

This produces a distribution over rotation angles that is biased toward smaller perturbations while still allowing large-angle corruption up to 90°, which corresponds to orthogonal tokens on the sphere. The motivation is twofold. First, a stronger mass near small angles makes the decoder spend more capacity on the local robustness regime that is most relevant to the prediction errors encountered during generation. Second, allowing occasional large-angle perturbations still provides a meaningful information bottleneck and prevents the decoder from overfitting to an excessively narrow neighborhood around the clean token. Under this corruption scheme, the tokenizer decoder is trained to reconstruct from randomly rotated noisy tokens. As a result, the decoder is explicitly encouraged to learn angular robustness on the spherical token space, without the need for an explicit spherical prior. In this sense, noise injection here serves both as a training-time robustness mechanism and as an implicit bottleneck that shapes the geometry of the learned token space.

3.3 Shaping per-token identifiability

As discussed above, the core idea behind identifiability is to emphasize more important content through higher-energy components. Here we aim to induce such a structure in the high-dimensional token space by explicitly introducing an availability bias over token dimensions and combining it with sufficiently strong spherical corruption. The resulting training dynamics encourage more frequently available dimensions to carry larger energy, thereby increasing their effective signal-to-noise ratio under corruption and making the token more identifiable.

Concretely, inspired by residual dropout strategies commonly used in RVQ-based systems [58], we introduce a postfix dimension dropout mechanism over the token dimensions. Let

$$x = (x_1, \ldots, x_N) \in \mathbb{S}^{N-1}(R) \subset \mathbb{R}^N$$

be the rotated noisy spherical token. For each token independently, with probability p we keep all dimensions unchanged. Otherwise, we sample an integer

$$K \sim \mathrm{Unif}\{1, 2, \ldots, N-1\},$$

and keep only the prefix dimensions $1, \ldots, K$, while dropping the postfix dimensions $K+1, \ldots, N$. Equivalently, if we define a random effective prefix length

$$K_{\mathrm{eff}} = \begin{cases} N, & \text{with probability } p, \\ K, & \text{with probability } 1-p, \end{cases}$$

then the dropout mask $m \in \{0,1\}^N$ is

$$m_i = \mathbf{1}[i \le K_{\mathrm{eff}}], \qquad i = 1, \ldots, N,$$

and the corrupted token is

$$x^{\mathrm{drop}} = x \odot m, \tag{10}$$

where $\odot$ denotes elementwise multiplication.

Under this sampling rule, when $p < 1$, lower-indexed dimensions have strictly higher probabilities of being retained. For any $i \in \{1, \ldots, N\}$, we have

$$\Pr(m_i = 1) = p + (1-p)\Pr(K \ge i) = p + (1-p)\frac{N-i}{N-1}, \tag{11}$$

where the last expression also gives $\Pr(m_1 = 1) = 1$ and $\Pr(m_N = 1) = p$. Therefore, this operation induces a clear prefix-to-postfix availability ordering over all dimensions. Note that the resulting token no longer lies on the high-dimensional sphere as we do not renormalize it before sending it to the decoder. We set $p = 0.5$ by default.

The training-dynamics intuition is straightforward. Because prefix dimensions are more likely to be preserved, the model is repeatedly required to reconstruct the signal from partially observed and corrupted tokens in which only a prefix, sometimes a short one, remains available. At the same time, the clean token has a fixed total energy budget due to the spherical token geometry. Under this budget, if the model is to maximize reconstruction robustness under such corruption, one natural optimization direction is to allocate larger energy to the more frequently available dimensions. In other words, the availability bias is transformed through training into an energy bias:

$$\text{higher availability} \implies \text{higher learned energy} \implies \text{higher corruption-time identifiability}.$$

3.4 Shaping a low-dimensional manifold

We would like to construct, at low cost, a low-dimensional manifold embedded in the high-dimensional token space that is both well-clustered and strongly interpolatable. Since the high-dimensional token lies on a sphere, a natural design is to introduce a low-dimensional spherical space and to align the two spaces using orthogonal projection and lifting operations. As discussed above, due to the spherical-cap covering behavior, different tokens overlap much more strongly under large rotational corruption in lower dimensions than in higher dimensions. Therefore, if we impose strong perturbations on the low-dimensional sphere, it naturally creates a much stronger information bottleneck, which in turn encourages stronger cross-token interpolatability through larger overlap regions. This is also reminiscent of the feature disentanglement and clustering effects induced by stronger bottlenecks such as those used in β-VAE [47].

To preserve geometric structure as much as possible, we use a learnable row-orthogonal linear projection to map the clean high-dimensional spherical token into a lower-dimensional spherical space. Let

$$x \in \mathbb{S}^{D_{\text{tok}}-1}(R_H)$$

be a native high-dimensional token, and let $d_{\text{core}} \ll D_{\text{tok}}$ denote the dimension of the core manifold. We use a learnable row-orthogonal projection matrix

$$W_\downarrow \in \mathbb{R}^{d_{\text{core}} \times D_{\text{tok}}}, \qquad W_\downarrow W_\downarrow^\top = I_{d_{\text{core}}},$$

to map the high-dimensional token into the low-dimensional spherical space. In implementation, we maintain an unconstrained matrix

$$A_\downarrow \in \mathbb{R}^{D_{\text{tok}} \times d_{\text{core}}},$$

compute its thin QR decomposition

$$A_\downarrow = Q_\downarrow R_\downarrow, \qquad Q_\downarrow^\top Q_\downarrow = I_{d_{\text{core}}},$$

and set

$$W_\downarrow = Q_\downarrow^\top.$$

We define the low-dimensional token as

$$z = R_L \frac{W_\downarrow x}{\|W_\downarrow x\|_2}, \qquad z \in \mathbb{S}^{d_{\text{core}}-1}(R_L). \tag{12}$$

The lifting map is defined by the same QR-based orthogonalization-and-renormalization template, but in the reverse direction. Specifically, we maintain an unconstrained matrix

$$A_\uparrow \in \mathbb{R}^{D_{\text{tok}} \times d_{\text{core}}},$$

compute its thin QR decomposition

$$A_\uparrow = Q_\uparrow R_\uparrow, \qquad Q_\uparrow^\top Q_\uparrow = I_{d_{\text{core}}},$$

and set

$$W_\uparrow = Q_\uparrow.$$

The lifted high-dimensional token is then defined as

$$x^{\text{lift}} = R_H \frac{W_\uparrow z}{\|W_\uparrow z\|_2}, \qquad x^{\text{lift}} \in \mathbb{S}^{D_{\text{tok}}-1}(R_H). \tag{13}$$

Since $W_\uparrow$ is column-orthogonal, $\|W_\uparrow z\|_2 = \|z\|_2 = R_L$ in exact arithmetic, so the above is equivalently

$$x^{\text{lift}} = \frac{R_H}{R_L} W_\uparrow z.$$

We keep the explicit renormalization in Eq. (13) for numerical consistency.

The motivation for using orthogonal mappings is geometric. The row-orthogonal projection $W_{\downarrow}$ has orthonormal rows and therefore acts as a partial isometry on its retained subspace; in particular, it does not introduce anisotropic scaling among the retained directions. The subsequent normalization maps the projected vector back onto the low-dimensional sphere. For the lifting step, the mapping is strictly angle-preserving on the low-dimensional sphere: for any two low-dimensional tokens $z_1, z_2 \in \mathbb{S}^{d-1}(R_L)$,

$$\frac{\langle W_{\uparrow} z_1, W_{\uparrow} z_2 \rangle}{\|W_{\uparrow} z_1\|_2 \, \|W_{\uparrow} z_2\|_2} = \frac{\langle z_1, z_2 \rangle}{\|z_1\|_2 \, \|z_2\|_2}. \tag{14}$$

Hence the angular similarity between low-dimensional tokens is preserved exactly after lifting.

To align the two spherical spaces, we employ a bidirectional commitment loss in the spirit of VQ-based discrete tokenizers. Let x denote the clean native high-dimensional token and x^{lift} the lifted low-dimensional token. We penalize their angular discrepancy in both directions using cosine similarity. Denoting stop-gradient by $\text{sg}[\cdot]$, we use the following bidirectional cosine commitment loss:

$$\mathcal{L}_{\text{commit}} = \big(1 - \cos(x, \text{sg}[x^{\text{lift}}])\big) + \big(1 - \cos(\text{sg}[x], x^{\text{lift}})\big). \tag{15}$$

This encourages the low-dimensional token to remain geometrically aligned with the high-dimensional token, while also encouraging the lifted low-dimensional representation to occupy a direction close to the original high-dimensional token.

We apply the strong information bottleneck and corruption to the low-dimensional sphere by constructing a noisy low-dimensional reconstruction path. That is, we apply the same spherical corruption process to the low-dimensional token z, lift the noisy low-dimensional token back to the high-dimensional sphere, and require the decoder to reconstruct the signal from this lifted noisy low-dimensional representation as well. If z^{rot} denotes the spherically corrupted low-dimensional token and x^{low} denotes its lifted high-dimensional version, then the decoder is trained on both reconstruction paths:

$$\text{corrupted native high-dimensional token} \;\rightarrow\; \text{decoder} \;\rightarrow\; \text{signal},$$

and

$$\text{lifted corrupted low-dimensional token} \;\rightarrow\; \text{decoder} \;\rightarrow\; \text{signal}.$$

As a result, the low-dimensional space is not merely an auxiliary projection of the high-dimensional token, but is explicitly required to support signal reconstruction under corruption, thereby encouraging it to encode the most reconstruction-critical information. Combined with the bidirectional commitment loss, this dual-path noisy reconstruction objective shapes a low-dimensional manifold with a strong information bottleneck and encourages the high-dimensional token space to be organized around the lifted low-dimensional geometry induced by the orthogonal mapping pair.

3.5 Bridge construction and training target in the generative model

For spherical tokens, the most natural evolution path is the geodesic path on the sphere, and the corresponding bridge construction is spherical flow matching (SFM), i.e., the spherical special case of Riemannian flow matching [19]. Several recent works have explored the use of SFM for improving generative modeling on spherical token spaces [60, 78]. We therefore use SFM as the default bridge and discuss several properties that motivate this choice.

Connection to VP-path flow. In high dimensions, SFM is closely related to the standard trigonometric interpolant bridge used in variance-preserving (VP) paths [1, 2, 94]. Let $x_1 \in \mathbb{S}^{D_{\text{tok}}-1}(R)$ denote a data token sampled from the tokenizer distribution. For the source endpoint, we sample an isotropic Gaussian vector $g \sim \mathcal{N}(0, I_{D_{\text{tok}}})$ and project it onto the same sphere:

$$x_0 = R\frac{g}{\|g\|_2}, \qquad x_0 \in \mathbb{S}^{D_{\text{tok}}-1}(R). \tag{16}$$

The VP-style trigonometric interpolation between the source sample x_0 and the target data token x_1 is

$$x_t^{\mathrm{VP}} = \cos\left(\frac{\pi t}{2}\right) x_0 + \sin\left(\frac{\pi t}{2}\right) x_1, \qquad t \in [0,1]. \tag{17}$$

Its squared norm is

$$\|x_t^{\mathrm{VP}}\|_2^2 = R^2 \cos^2\left(\frac{\pi t}{2}\right) + R^2 \sin^2\left(\frac{\pi t}{2}\right) + 2\cos\left(\frac{\pi t}{2}\right)\sin\left(\frac{\pi t}{2}\right)\langle x_0, x_1\rangle. \tag{18}$$

Equivalently,

$$\|x_t^{\mathrm{VP}}\|_2^2 = R^2 + 2\cos\left(\frac{\pi t}{2}\right)\sin\left(\frac{\pi t}{2}\right)\langle x_0, x_1\rangle.$$

Conditioned on any fixed data token x_1, the source direction x_0/R is uniformly distributed on the unit sphere. Therefore,

$$\mathbb{E}\left[\frac{\langle x_0, x_1\rangle}{R^2}\,\middle|\, x_1\right] = 0, \qquad \frac{\langle x_0, x_1\rangle}{R^2} = O_p\left(D_{\mathrm{tok}}^{-1/2}\right).$$

Hence, in high dimensions,

$$\frac{\langle x_0, x_1\rangle}{R^2} \approx 0.$$

It follows that

$$\|x_t^{\mathrm{VP}}\|_2^2 \approx R^2,$$

which shows that the VP trigonometric path is approximately norm-preserving. Under the same high-dimensional near-orthogonality condition, the geodesic angle

$$\Omega = \arccos\left(\frac{\langle x_0, x_1\rangle}{R^2}\right)$$

concentrates near $\pi/2$, and the SFM geodesic interpolation

$$x_t^{\mathrm{SFM}} = \frac{\sin((1-t)\Omega)}{\sin\Omega} x_0 + \frac{\sin(t\Omega)}{\sin\Omega} x_1 \tag{19}$$

reduces approximately to

$$x_t^{\mathrm{SFM}} \approx \cos\left(\frac{\pi t}{2}\right) x_0 + \sin\left(\frac{\pi t}{2}\right) x_1. \tag{20}$$

Therefore, in the high-dimensional near-orthogonal regime, SFM is nearly equivalent to the VP-style trigonometric interpolant on the same radius-R sphere.

Norm stability of x_t. A further property of SFM is that the norm of x_t remains constant along the bridge. By contrast, under a linear interpolation bridge $x_t = (1-t)x_0 + tx_1$, the norm of x_t varies with t. In the high-dimensional near-orthogonal case, for example, we have $\|x_t\|_2^2 \approx R^2\big((1-t)^2 + t^2\big)$, which is strongly coupled with the bridge time. Since the model typically receives an explicit time embedding, such a bridge introduces an additional and unnecessary coupling between time and radial scale. At inference time, once the generated trajectory deviates from the ideal bridge, the radial scale of the current state may become inconsistent with the explicit time embedding, increasing the risk of exposure bias.

Equivalence of x-pred and v-pred. As discussed above, in raw signal-space generative modeling, the manifold hypothesis often motivates direct target prediction, which can be more effective than predicting noise or velocity in certain regimes. It is useful to first contrast the role of endpoint prediction in Euclidean rectified flow and in spherical flow matching. For the standard Euclidean rectified-flow bridge

$$x_t = (1-t)x_0 + tx_1, \qquad t \in [0,1],$$

the oracle velocity is $v_t = x_1 - x_0$, which lives in the full ambient Euclidean space and is unconstrained. Therefore, a direct v-pred parameterization requires the model output head to predict a vector containing the high-entropy source term x_0. By contrast, an x-pred parameterization lets the model predict $\hat{x}_1$ and converts it into a velocity by

$$\hat{v}_t^{\mathrm{RF}} = \frac{\hat{x}_1 - x_t}{1-t}. \tag{21}$$

Algebraically, x-pred and v-pred can be converted into each other in the Euclidean bridge. However, as discussed in recent works [63], they are not equivalent as output-layer parameterizations: x-pred asks the model to predict the data endpoint, which may benefit from the manifold structure of the data distribution, whereas v-pred directly asks the model to predict the unconstrained displacement $x_1 - x_0$.

For spherical flow matching, the situation is different. A valid velocity at the bridge state $x_t \in \mathbb{S}^{D_{\mathrm{tok}}-1}(R)$ must lie in the tangent space

$$T_{x_t}\mathbb{S}^{D_{\mathrm{tok}}-1}(R) = \left\{ u \in \mathbb{R}^{D_{\mathrm{tok}}} : \langle u, x_t \rangle = 0 \right\}.$$

Thus, unlike Euclidean rectified flow, the velocity is not an unconstrained ambient vector, and any direct velocity parameterization must either explicitly predict a tangent vector or project an ambient prediction onto the tangent space. For any $y \in \mathbb{S}^{D_{\mathrm{tok}}-1}(R)$, define the geodesic angle between x_t and y as

$$\alpha_t(y) = \arccos\left(\frac{\langle x_t, y \rangle}{R^2} \right).$$

The tangent projection of y at x_t is

$$\Pi_{x_t}^{\perp}(y) = y - \frac{\langle y, x_t \rangle}{R^2} x_t = y - \cos\big(\alpha_t(y)\big) x_t. \tag{22}$$

The Riemannian logarithm map from x_t to y is

$$\log_{x_t}(y) = \frac{\alpha_t(y)}{\sin \alpha_t(y)} \Pi_{x_t}^{\perp}(y), \qquad \log_{x_t}(y) \in T_{x_t}\mathbb{S}^{D_{\mathrm{tok}}-1}(R), \tag{23}$$

with

$$\| \log_{x_t}(y) \|_2 = R\, \alpha_t(y).$$

Therefore, if the model predicts a spherical endpoint $\hat{x}_1$, the corresponding valid SFM velocity over the remaining interval $[t, 1]$ is

$$\hat{v}_t^{\mathrm{full}} = \frac{1}{1-t} \log_{x_t}(\hat{x}_1). \tag{24}$$

The oracle SFM velocity is obtained by setting $y = x_1$. Let

$$\Omega = \arccos\left(\frac{\langle x_0, x_1 \rangle}{R^2} \right),$$

and let the SFM bridge be

$$x_t = \frac{\sin((1-t)\Omega)}{\sin \Omega} x_0 + \frac{\sin(t\Omega)}{\sin \Omega} x_1. \tag{25}$$

Then the oracle tangent velocity is

$$v_t = \dot{x}_t = \frac{\Omega}{\sin \Omega} \left(-\cos((1-t)\Omega) x_0 + \cos(t\Omega) x_1 \right), \tag{26}$$

or equivalently

$$v_t = \frac{1}{1-t} \log_{x_t}(x_1).$$

Thus, for SFM, estimating a geometrically valid velocity is essentially equivalent to estimating a spherical endpoint together with its induced tangent direction and scale. In this sense, the distinction between x-pred and v-pred is weaker than in Euclidean rectified flow: the tangent-space constraint naturally turns a valid velocity parameterization into an endpoint-induced parameterization.

Under-stepping and direction-only modeling. A conventional velocity objective would minimize the full velocity MSE,

$$\mathcal{L}_{\mathrm{v}} = \|\hat{v}_t - v_t\|_2^2. \tag{27}$$

However, in the present high-dimensional spherical setting, full velocity MSE introduces a potential under-stepping effect. To see this, suppose the oracle velocity can be written as

$$v_t = s_t d_t, \qquad s_t = \|v_t\|_2, \qquad \|d_t\|_2 = 1,$$

and suppose the model predicts a tangent velocity but with a direction error. Write its prediction as

$$\hat{v}_t = a_t \hat{d}_t, \qquad \|\hat{d}_t\|_2 = 1,$$

where $a_t \geq 0$ is the predicted speed. Let ψ_t denote the angle between the predicted and oracle directions:

$$\cos\psi_t = \langle \hat{d}_t, d_t \rangle.$$

For a fixed predicted direction $\hat{d}_t$, the velocity MSE as a function of the predicted magnitude a_t is

$$\begin{aligned} \left\| a_t \hat{d}_t - s_t d_t \right\|_2^2 &= a_t^2 - 2a_t s_t \langle \hat{d}_t, d_t \rangle + s_t^2 \\ &= a_t^2 - 2a_t s_t \cos\psi_t + s_t^2. \end{aligned} \tag{28}$$

Minimizing this quadratic over a_t gives

$$a_t^\star = s_t \cos\psi_t, \tag{29}$$

Therefore, unless the predicted direction is perfectly aligned with the oracle direction, the nonnegative MSE-optimal velocity magnitude is smaller than the oracle magnitude. In other words, full velocity MSE allows the model to explain directional uncertainty by reducing the step size.

For the SFM geodesic, the oracle speed is constant along the bridge and satisfies exactly

$$\|v_t\|_2 = R\Omega. \tag{30}$$

In high dimensions, when x_0/R and x_1/R are weakly correlated random unit directions, their normalized inner product concentrates near zero, and hence

$$\Omega \approx \frac{\pi}{2}.$$

Thus the oracle velocity norm concentrates near

$$\|v_t\|_2 \approx \frac{\pi}{2} R. \tag{31}$$

Under full velocity MSE, however, if the model has high uncertainty in endpoint or direction estimation, Eq. (29) encourages a smaller predicted speed. During inference, this can lead to a trajectory whose accumulated path length is systematically shorter than the typical SFM geodesic length. In the high-dimensional regime, where the target endpoint is typically close to 90° away from the source endpoint, such under-stepping creates the risk that the generated endpoint remains at an angle substantially smaller than 90° from the source sample. To avoid this failure mode, we use a direction-only parameterization: the model still predicts a spherical endpoint $\hat{x}_1$, but this endpoint is used only to define the tangent direction at the current bridge state. Specifically, we define the predicted and oracle unit tangent directions as

$$\hat{d}_t = \frac{\log_{x_t}(\hat{x}_1)}{\|\log_{x_t}(\hat{x}_1)\|_2}, \qquad d_t = \frac{v_t}{\|v_t\|_2} = \frac{\log_{x_t}(x_1)}{\|\log_{x_t}(x_1)\|_2}. \tag{32}$$

We then supervise only the tangent direction using the cosine direction loss

$$\mathcal{L}_{\mathrm{dir}} = 1 - \langle \hat{d}_t, d_t \rangle. \tag{33}$$

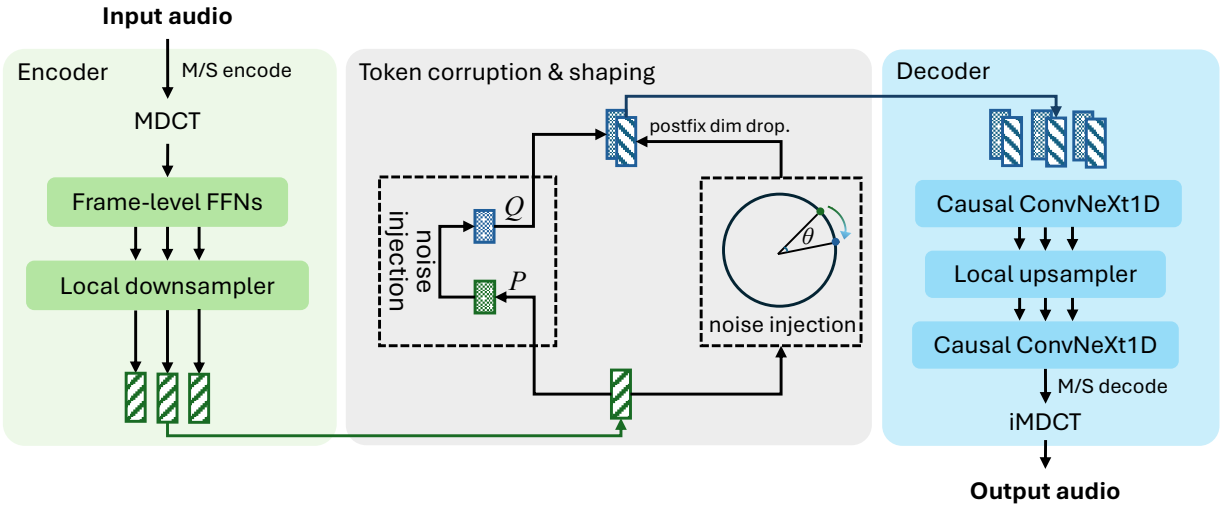

Figure 1 Illustration of the Locodec tokenizer architecture, consisting of MDCT coefficient extraction, a fully local encoder, a token-grid refiner, a coefficient-grid predictor, and an inverse-MDCT stage for waveform reconstruction.

At inference time, before any additional guidance combination, we fix the predicted velocity magnitude to its high-dimensional SFM limit:

$$\hat{v}_t = \frac{\pi R}{2}\hat{d}_t. \tag{34}$$

In this way, the model-predicted endpoint serves as a direction parameterization rather than as an exact endpoint estimate or a full velocity estimate. This removes the need for the model to predict the velocity norm, avoids MSE-induced under-stepping, and keeps the generated trajectory length compatible with the typical high-dimensional SFM geometry.

4 Model Architecture Design

In this section, we provide concrete architecture instantiations of a minimalist tokenizer model and a minimalist AR flow-matching model. For the tokenizer, we intentionally restrict ourselves to simple and standard building blocks, in order to show that once the token space is shaped appropriately, neither the overall model structure nor its basic units need to be overly complicated to achieve a good balance between fidelity and predictability. For the AR flow-matching model, we adopt a modular design that is tailored to the error-accumulation and exposure-bias issues that arise easily in AR flow systems. Through explicit functional disentanglement, the system can, via training dynamics, automatically optimize different information extraction paths, while at inference time appropriate multi-path CFG can be used to selectively enhance different functional components, thereby mitigating sequential error accumulation during AR generation.

4.1 Locodec: a minimalist, locally encoded tokenizer architecture

We propose Locodec, a locally encoded codec. Our goal here is deliberately minimalist: the architecture is built from simple, standard components, so that the emphasis remains on token-space shaping rather than on architectural complexity. Moreover, although the experiments in this paper mainly focus on single-channel TTS, the tokenizer design and training losses described here are formulated to support both monaural and stereo audio in a unified manner. Figure 1 shows the Locodec pipeline.

Mid/side input formulation and MDCT front-end. We adopt a mid/side formulation at the waveform level. For the left and right channels x_L and x_R, we define

$$x_{\text{mid}} = \frac{x_L + x_R}{2}, \qquad x_{\text{side}} = \frac{x_L - x_R}{2}. \tag{35}$$

For monaural input, we simply set $x_L = x_R$ so that $x_{\text{side}} = 0$. This allows monaural and stereo samples to be mixed naturally during training while sharing the same tokenizer architecture and loss functions.

Motivated by conventional audio codecs [8, 13, 91], we apply an MDCT transform to obtain real-valued time–frequency representations, i.e., MDCT coefficients, for both the mid and side waveforms. For an MDCT window size of k, the corresponding hop size is $k/2$, which determines the native coefficient-frame rate seen by the tokenizer encoder. We further apply a simple signed dynamic-range compression to both the mid and side coefficient sequences before feeding them into the encoder:

$$c \leftarrow \operatorname{sign}(c)\,|c|^{1/3}. \tag{36}$$

This transformation preserves coefficient sign while compressing amplitude range, reducing the tendency of the model to over-emphasize large low-frequency coefficients and underfit smaller but still perceptually important higher-frequency components. The compressed mid and side coefficient sequences are then concatenated along the feature dimension and used as the input to the tokenizer encoder.

Fully local encoder. On top of this MDCT coefficient sequence, we use a fully local encoder to map the coefficients to a hidden space. The encoder first applies a stack of frame-level local feed-forward networks (FFNs) at the native MDCT frame rate, and then uses a single-step linear downsampling layer that groups s neighboring frame-level embeddings, concatenates them along the feature dimension, and projects the concatenated vector directly into the target token dimension. More precisely, if the frame-level embedding dimension is D_{enc} and the temporal downsampling factor is s, then the downsampling layer is implemented as a linear map

$$\mathbb{R}^{D_{\text{enc}} \cdot s} \to \mathbb{R}^{D_{\text{tok}}}.$$

The token sequence is strictly local in the sense that each token is formed only from its corresponding local coefficient frames, without cross-token information exchange. We intentionally keep the encoder fully local to make the tokenization closer in spirit to raw input-space modeling with non-overlapping chunking or patchification: each token corresponds to a local region of the input signal, and the corresponding manifold assumption is imposed at the level of local signal chunks. Moreover, this locality also reduces the risk of reconstruction conflicts during generation. If the encoder introduces strong cross-token interactions, then the information required to reconstruct a given signal segment may be distributed across multiple neighboring tokens. While the tokenizer decoder can exploit such distributed information during reconstruction training, an AR generative model predicts these tokens separately at inference time, and the prediction errors or inconsistencies among neighboring tokens may then provide conflicting evidence about the same underlying signal segment, making reconstruction less stable. We therefore isolate the information scope of each token at the encoder level, so that each token primarily models its own local signal region.

Each token is then normalized onto the high-dimensional sphere. As described in the previous section, all token-space shaping mechanisms, including spherical corruption, postfix dimension dropout, low-dimensional projection and lifting, and dual-path noisy reconstruction, are applied on top of this token sequence.

Stacked causal convolutional decoder. The decoder starts from either a noisy high-dimensional token sequence or a lifted noisy low-dimensional token sequence. It first applies a token-grid refinement module implemented as stacked causal ConvNeXt1D blocks [69]. This stage operates directly on the low-frame-rate token sequence and is responsible for reconciling the corrupted token representation before coefficient-frame reconstruction. The refined token sequence is then mapped back to the native MDCT coefficient-frame rate through a single-step linear upsampling layer, which serves as the decoder-side counterpart of the single-step linear downsampling layer. Another stack of causal ConvNeXt1D blocks then performs coefficient-grid prediction at the native MDCT frame rate to recover frame-level detail. The mid/side hidden representations are then decoded via a channel-separation layer: let $h \in \mathbb{R}^{T_{\text{coef}} \times D}$ denote the refined hidden sequence produced by the coefficient-grid ConvNeXt1D predictor, where T_{coef} is the native MDCT coefficient-frame length and D is the hidden dimension. We pass h through two separate FFNs to obtain a mid-channel hidden sequence and a side-channel hidden sequence:

$$h_{\text{mid}} = \text{FFN}_{\text{mid}}(h), \qquad h_{\text{side}} = \text{FFN}_{\text{side}}(h). \tag{37}$$

These two hidden sequences are then mapped to coefficient predictions using a shared gated output layer: let $\mathcal{O}_{\text{coef}} : \mathbb{R}^D \to \mathbb{R}^{2F}$ denote the shared coefficient output layer, where F is the number of MDCT frequency bins per frame. Applying $\mathcal{O}_{\text{coef}}$ to either h_{mid} or h_{side} produces two tensors, corresponding to an amplitude branch and a signed-gate branch. Writing

$$(h^{(b)}_{\text{amp}}, h^{(b)}_{\text{sgn}}) = \mathcal{O}_{\text{coef}}(h_b), \qquad b \in \{\text{mid}, \text{side}\}, \tag{38}$$

the predicted MDCT coefficients for branch b are parameterized as

$$\widehat{c}_b \;=\; \exp\big(h^{(b)}_{\text{amp}}\big) \cdot \tanh\big(h^{(b)}_{\text{sgn}}\big), \qquad b \in \{\text{mid}, \text{side}\}. \tag{39}$$

Here $\exp(h^{(b)}_{\text{amp}})$ controls a nonnegative amplitude scale, while $\tanh(h^{(b)}_{\text{sgn}})$ acts as a bounded signed gate. The left and right MDCT coefficients are recovered by

$$\widehat{c}_L = \widehat{c}_{\text{mid}} + \widehat{c}_{\text{side}}, \qquad \widehat{c}_R = \widehat{c}_{\text{mid}} - \widehat{c}_{\text{side}}. \tag{40}$$

Finally, inverse MDCT is applied to $\widehat{c}_L$ and $\widehat{c}_R$ to reconstruct the left and right waveforms $\hat{x}_L$ and $\hat{x}_R$ in the original time domain.

Training objectives. The training objective consists of three parts: a noisy reconstruction loss, a perceptual loss, and the high–low-dimensional commitment loss.

The noisy reconstruction loss is applied to both high-dimensional and low-dimensional reconstruction paths. Following prior works [58, 72], we compute STFT magnitudes at window sizes

$$\mathcal{W}_{\text{STFT}} = \{2^5, 2^6, \ldots, 2^{12}\}.$$

For each $w \in \mathcal{W}_{\text{STFT}}$, define

$$A_w(y) = \left|\text{STFT}_w(y)\right|.$$

Here and below, y denotes a waveform and $\hat{y}$ denotes its reconstruction. The normalized magnitude loss is defined channel-wise. For the left, right, and mid channels, we use the target magnitude of the same channel as the normalization factor:

$$r^b_w(y) = \text{mean}(A_w(y_b)), \qquad b \in \{L, R, \text{mid}\}. \tag{41}$$

Here mean$(\cdot)$ averages over the time–frequency bins of a single example. For the side channel, however, we normalize by the target mid-channel magnitude:

$$r^{\text{side}}_w(y) = \text{mean}(A_w(y_{\text{mid}})). \tag{42}$$

This avoids an ill-conditioned normalization in the monaural case, where the target side channel is identically zero. The channel-wise normalized magnitude loss is then

$$\ell^{\text{mag},b}_w(\hat{y}, y) = \mathbb{E}\left[\frac{\text{mean}(|A_w(\hat{y}_b) - A_w(y_b)|)}{r^b_w(y) + \varepsilon}\right], \qquad b \in \{L, R, \text{mid}, \text{side}\}, \tag{43}$$

where $\mathbb{E}[\cdot]$ denotes the empirical average over the training batch. The log-magnitude loss is applied only to the left and right channels:

$$\ell^{\text{lmag},b}_w(\hat{y}, y) = \mathbb{E}\left[\text{mean}\big(\big|\log_{10}\big(A_w(\hat{y}_b) + \varepsilon\big) - \log_{10}\big(A_w(y_b) + \varepsilon\big)\big|\big)\right], \qquad b \in \{L, R\}. \tag{44}$$

For a reconstructed stereo waveform, we define

$$\hat{y}_{\text{mid}} = \frac{\hat{y}_L + \hat{y}_R}{2}, \qquad \hat{y}_{\text{side}} = \frac{\hat{y}_L - \hat{y}_R}{2},$$

and analogously for the target waveform y. Let $\hat{y}^{(H)}$ and $\hat{y}^{(L)}$ denote the reconstructions from the high-dimensional and low-dimensional paths, respectively. We define the reconstruction loss for each path as

$$\mathcal{L}_{\text{rec}}^{(m)} = \sum_{w \in \mathcal{W}_{\text{STFT}}} \Bigg[\sum_{b \in \{L,R\}} \Big(\ell_w^{\text{mag},b}(\hat{y}^{(m)}, y) + \lambda_{\text{lmag}} \ell_w^{\text{lmag},b}(\hat{y}^{(m)}, y) \Big) + \ell_w^{\text{mag,mid}}(\hat{y}^{(m)}, y) + \ell_w^{\text{mag,side}}(\hat{y}^{(m)}, y) \Bigg], \qquad m \in \{H, L\}. \tag{45}$$

In the current implementation, we use the same reconstruction-loss form for both paths, while the low-dimensional path is assigned a smaller weight in the final objective.

The perceptual loss used to train the tokenizer decoder is composed of an adversarial generator loss and a feature-matching loss. The GAN discriminator operates on multi-resolution uncompressed MDCT coefficients computed directly from the waveform, and is built from stacked Conv1d blocks [72]. We use MDCT coefficient matrices computed with four window sizes 256, 512, 1024, 2048, and for each resolution, we instantiate both a full-band discriminator and a sub-band discriminator. The full-band discriminator takes the entire coefficient matrix as input, and the sub-band discriminator uniformly partitions the spectrum into sub-bands of 4-kHz bandwidth. In implementation, the sub-band discriminator is realized with grouped convolutions: each spectral sub-band is processed by its own convolutional group, while all sub-bands at the same MDCT resolution are treated as one discriminator module. Therefore, each resolution contributes one full-band discriminator and one grouped sub-band discriminator, yielding a total of eight discriminator modules across the four resolutions.

We use the least-squares GAN (LSGAN) objective [74]. Let $\mathcal{J}$ denote the set of all discriminators. For each $j \in \mathcal{J}$, let $C_j(\cdot)$ denote the MDCT representation associated with discriminator D_j. The discriminator loss is

$$\mathcal{L}_D^{(j)} = \mathbb{E}_y \left[\left(D_j(C_j(y)) - 1 \right)^2 \right] + \mathbb{E}_{\hat{y}^{(H)}} \left[D_j\Big(C_j(\text{sg}[\hat{y}^{(H)}]) \Big)^2 \right], \tag{46}$$

and the adversarial generator loss for the tokenizer is

$$\mathcal{L}_{\text{adv}}^{(m,j)} = \mathbb{E}_{\hat{y}^{(m)}} \left[\Big(D_j(C_j(\hat{y}^{(m)})) - 1 \Big)^2 \right], \quad m \in \{H, L\}. \tag{47}$$

Note that the discriminators are trained using only the high-dimensional reconstructions as negative samples.

A feature-matching loss is also computed as a layer-wise normalized MAE between the discriminator hidden activations of the reconstructed and target waveforms. Let $D_j^{(\ell)}(\cdot)$ denote the hidden activation of the ℓ-th layer of discriminator D_j, and let L_j be the number of hidden layers used for feature matching. We define

$$\mathcal{L}_{\text{FM}}^{(m,j)} = \frac{1}{L_j} \sum_{\ell=1}^{L_j} \mathbb{E} \left[\frac{\text{mean}\Big(\Big| D_j^{(\ell)}(C_j(\hat{y}^{(m)})) - \text{sg}\Big[D_j^{(\ell)}(C_j(y)) \Big] \Big| \Big)}{\text{mean}\Big(\Big| \text{sg}\Big[D_j^{(\ell)}(C_j(y)) \Big] \Big| \Big) + \epsilon} \right], \quad m \in \{H, L\}. \tag{48}$$

The overall perceptual loss used to train the tokenizer is

$$\mathcal{L}_{\text{perc}}^{(m)} = \sum_{j \in \mathcal{J}} \left(\mathcal{L}_{\text{adv}}^{(m,j)} + \mathcal{L}_{\text{FM}}^{(m,j)} \right), \quad m \in \{H, L\}, \tag{49}$$

and the discriminator objective is

$$\mathcal{L}_D = \sum_{j \in \mathcal{J}} \mathcal{L}_D^{(j)}. \tag{50}$$

Putting everything together, the overall tokenizer objective takes the form

$$\mathcal{L}_{\text{tok}} = \mathcal{L}_{\text{rec}}^{(H)} + \mathcal{L}_{\text{perc}}^{(H)} + \lambda_{\text{low}} \left(\mathcal{L}_{\text{rec}}^{(L)} + \mathcal{L}_{\text{perc}}^{(L)} \right) + \lambda_{\text{commit}} \mathcal{L}_{\text{commit}}, \tag{51}$$

where $\mathcal{L}_{\text{commit}}$ is the bidirectional high–low-dimensional commitment loss defined previously. We set $\lambda_{\text{lmag}} = 0.2$, $\lambda_{\text{low}} = 0.2$, and $\lambda_{\text{commit}} = 0.1$ by default.

4.2 MP-ELD: a multi-path information-routing encoder-LM-decoder architecture

We find that constraining tokens to a high-dimensional sphere or explicitly learning a low-dimensional manifold does not automatically solve the AR error-accumulation problem. Based on empirical observations, we hypothesize that one important remaining source of AR error accumulation is information-pathway conflict under CFG. Specifically, conditioning signals with partially overlapping functionality may appear in multiple CFG paths but be combined inconsistently across paths; the resulting conflicts are then fed back through the AR loop and manifest as gradual drift of the corresponding attributes. In practice, degradation in a TTS system often emerges as drift in acoustic attributes—e.g., spectral distortion, gradually drifting loudness, or changes in speaking rate—whereas content consistency is typically much more stable. This asymmetry suggests that acoustic-state drift, rather than content inconsistency, is the dominant mode of error accumulation. We attribute it to partially overlapping or redundant acoustic cues across CFG paths that can become mutually inconsistent under guidance, thereby inducing systematic drift in the AR inference process. Motivated by this hypothesis, we explicitly decouple information pathways by routing conditioning information into functionally distinct channels and applying multi-path CFG as a structured residual correction, so that each guidance component targets a specific role rather than implicitly entangling overlapping information across paths.

The ELD framework. We start from the ELD (Encoder–LM–Decoder) framework, which is a common recipe for in-context conditional AR generation [53, 54, 61, 89, 116]. Given a user-specified or task-provided control signal c (e.g., text, labels, or reference inputs), ELD aggregates the control signal c and the native-token prefix $x_{1:i}$ into a step-wise conditioning signal c'_i, and the decoder predicts the next native high-dimensional token conditioned on c'_i:

$$p_\theta(x_{i+1} \mid x_{1:i}, c) = p_\theta(x_{i+1} \mid c'_i), \qquad c'_i = f_\theta(x_{1:i}, c). \tag{52}$$

ELD constructs c'_i using an encoder, a sequence modeling module (LM), and a flow-matching decoder. A token encoder E maps a clean native token to a hidden representation:

$$h_i = E(x_i), \qquad h_i \in \mathbb{R}^{D_{\text{model}}}. \tag{53}$$

This serves as a token adapter that maps the token into a hidden space better aligned with sequential modeling and conditioning. The control signal c is embedded as a sequence $\phi(c)$ in the same hidden space and concatenated with the encoded prefix:

$$s_i = [\,\phi(c)\,;\, h_{1:i}\,], \tag{54}$$

and an LM then produces step-wise conditioning vectors

$$c'_{1:i} = \mathrm{LM}(s_i), \qquad c'_j \in \mathbb{R}^{D_{\text{model}}}. \tag{55}$$

Given c'_i and the bridge state at position $i+1$, the decoder predicts a spherical endpoint estimate $\hat{x}_{i+1}$. This endpoint estimate is then converted, using the logarithm-map construction described in the previous section, into a pathwise fixed-norm tangent velocity estimate before CFG. In addition to the step-wise conditioning vector, the decoder also receives the bridge time $\tau \in [0, 1]$ and its time embedding.

To mitigate exposure bias, during training the ground-truth prefix tokens are perturbed with small noise before being fed to the encoder. We denote the resulting noisy teacher-forced native token by

$$x_i^{\text{ctx}} = \mathrm{Corrupt}(x_i).$$

This simulates mild inference-time distribution drift while staying within a locally reconstruction-stable neighborhood and is a standard technique [17, 82]. At inference time, x_i^{ctx} denotes the previously generated native token at step i. Moreover, CFG is applied only at the decoder level, while the encoder and LM are evaluated once to produce the step-wise conditioning vectors. This is computationally cheaper than settings in which the LM or backbone must participate in CFG, since guidance does not require multiple forward passes through the encoder or LM.

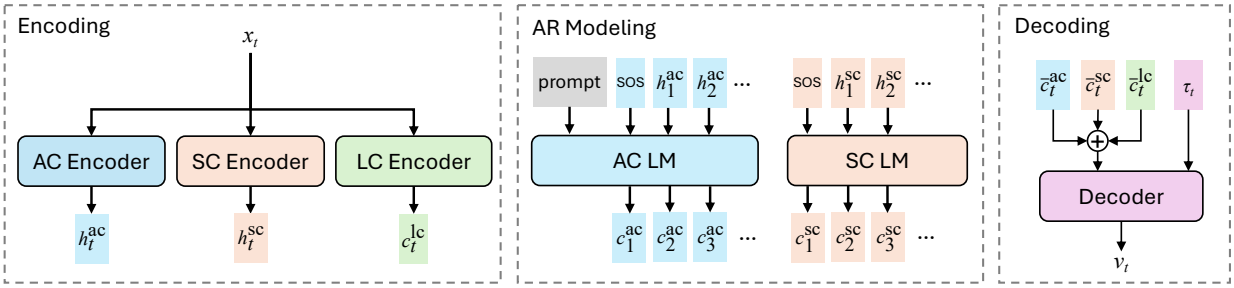

Figure 2 Illustration of the MP-ELD framework. Multiple local encoders map the same input token into distinct hidden spaces, whose functional roles are shaped by different AR information paths through training dynamics. A flow-matching decoder predicts pathwise next-token velocity fields conditioned on the corresponding path embeddings, which are combined by multi-path residual CFG during inference.

However, in practice we find that the ELD framework can become fragile when stronger CFG is applied in the decoder. Empirically, this failure mode is closely tied to two design ambiguities: (i) what information the step-wise conditioning signal is expected to preserve and how it should be used or amplified by the decoder, and (ii) how to define the "conditional" and "unconditional" paths in CFG for the decoder. In particular, a fully null unconditional path can be problematic in this setting, since it provides no local continuation anchor and may yield a high-variance early-time velocity estimate, making training statistically difficult. Based on these observations, we introduce multi-path ELD (MP-ELD), a modification of ELD that makes the roles of conditioning signals explicit. The core idea is information routing: we construct multiple conditioning paths that intentionally receive different information, so that each path naturally specializes to a different role. Figure 2 illustrates the MP-ELD framework.

At AR step i, MP-ELD constructs three step-wise conditioning vectors:

- Local-continuity conditioning c_i^{lc}: a local continuation anchor derived from the current token, mainly responsible for enforcing short-range continuity.
- Self-consistency conditioning c_i^{sc}: an in-context summary of modality-internal evolution over the prefix, mainly responsible for long-horizon stability and internal attribute consistency.
- Alignment-consistency conditioning c_i^{ac}: a condition-aware signal mainly responsible for aligning the generated output with the external control signal.

All three are represented in a shared D_{model}-dimensional conditioning space. For notational clarity, we use $\bar{c}_i^{(\cdot)}$ to denote the decoder-side version of each conditioning vector, and write a path configuration as a tuple only to indicate which components are present.

Information modeling paths. We construct three conditioning sources using three lightweight token encoders and two LMs:

- A local encoder E_{lc} produces the local-continuity signal from the current token:

$$c_i^{\mathrm{lc}} = E_{\mathrm{lc}}(x_i^{\mathrm{ctx}}) \in \mathbb{R}^{D_{\mathrm{model}}}. \tag{56}$$

  This can be viewed as a local-continuation path with minimal self-conditioning.

- A self-consistency encoder E_{sc} and a self-consistency LM $\mathrm{LM}_{\mathrm{sc}}$ produce the self-consistency signal from the history tokens:

$$h_i^{\mathrm{sc}} = E_{\mathrm{sc}}(x_i^{\mathrm{ctx}}), \qquad c_{1:i}^{\mathrm{sc}} = \mathrm{LM}_{\mathrm{sc}}(h_{1:i}^{\mathrm{sc}}), \qquad c_i^{\mathrm{sc}} \in \mathbb{R}^{D_{\mathrm{model}}}. \tag{57}$$

  This can be viewed as a global-continuation path with full self-conditioning over the token history.

- An alignment-consistency encoder E_{ac} and an alignment-consistency LM LM_{ac} produce the alignment signal by integrating the external condition with the history tokens:

$$h_i^{\text{ac}} = E_{\text{ac}}(x_i^{\text{ctx}}), \qquad c_{1:i}^{\text{ac}} = \text{LM}_{\text{ac}}\big([\phi(c);\, h_{1:i}^{\text{ac}}]\big), \qquad c_i^{\text{ac}} \in \mathbb{R}^{D_{\text{model}}}. \tag{58}$$

  This can be viewed as a cross-modal, externally conditioned global-continuation path.

These three paths can be intuitively interpreted as follows:

- c_i^{lc} encourages smooth local continuation of short-range acoustic attributes, such as pitch, phase, and energy, and helps prevent abrupt discontinuities.
- c_i^{sc} captures slowly varying acoustic attributes that should remain stable over an utterance, such as timbre, overall energy profile, and accent or style.
- c_i^{ac} indicates where the model is in the conditioned content and therefore what should be generated next, e.g., which part of the text should be spoken at the current step.

We augment the alignment-consistency LM with an additional scalar head for stop detection. Concretely, in addition to the step-wise alignment vector c_i^{ac}, the alignment-consistency LM outputs a scalar

$$\pi_i \in [0, 1]$$

representing the probability of continuing generation after step i. During training, we supervise π_i with a binary cross-entropy loss against the ground-truth continuation label derived from the sequence length. During inference, we terminate decoding when $\pi_i < 0.5$.

Orthogonal residual conditioning. To encourage a residual, non-overlapping decomposition of conditioning information, we orthogonalize the three condition vectors via a Gram–Schmidt transform and normalize each component to a fixed radius. For a target radius $r = \sqrt{D_{\text{model}}}$, let

$$\text{Norm}_r(u) = r\frac{u}{\|u\|_2 + \epsilon}.$$

At AR step i, we construct the decoder-side conditioning components as

$$\bar{c}_i^{\text{lc}} = \text{Norm}_{\sqrt{D_{\text{model}}}}\left(c_i^{\text{lc}}\right), \tag{59}$$

$$\bar{c}_i^{\text{sc}} = \text{Norm}_{\sqrt{D_{\text{model}}}}\left(c_i^{\text{sc}} - \frac{\langle c_i^{\text{sc}}, \bar{c}_i^{\text{lc}}\rangle}{\|\bar{c}_i^{\text{lc}}\|_2^2 + \epsilon}\bar{c}_i^{\text{lc}}\right), \tag{60}$$

$$\bar{c}_i^{\text{ac}} = \text{Norm}_{\sqrt{D_{\text{model}}}}\left(c_i^{\text{ac}} - \frac{\langle c_i^{\text{ac}}, \bar{c}_i^{\text{lc}}\rangle}{\|\bar{c}_i^{\text{lc}}\|_2^2 + \epsilon}\bar{c}_i^{\text{lc}} - \frac{\langle c_i^{\text{ac}}, \bar{c}_i^{\text{sc}}\rangle}{\|\bar{c}_i^{\text{sc}}\|_2^2 + \epsilon}\bar{c}_i^{\text{sc}}\right). \tag{61}$$

Intuitively, this orthogonalization operation encourages each condition vector to encode residual information that the preceding condition vectors do not contain. This allows us to define the full-path condition as the summation of the orthogonalized components instead of the concatenation of the original vectors, which also controls the overall conditioning dimension and the related model complexity when D_{model} is large.

Multi-path residual CFG. Using the orthogonalized components defined above, we apply CFG dropout only to the non-local branches, while always keeping the local-continuity anchor $\bar{c}_i^{\text{lc}}$ present. This preserves an always-available local continuation signal throughout training. For a given path configuration, we form the additive path condition

$$\tilde{c}_i = \bar{c}_i^{\text{lc}} + \delta_i^{\text{sc}}\bar{c}_i^{\text{sc}} + \delta_i^{\text{ac}}\bar{c}_i^{\text{ac}}, \qquad \delta_i^{\text{sc}}, \delta_i^{\text{ac}} \in \{0, 1\}. \tag{62}$$

The decoder receives the concatenation of the additive path condition and the bridge-time embedding $e_{\text{time}}(\tau)$, denoted by $q_i(\tau)$. We use three path configurations in this work:

- L: local-continuity only,

$$\tilde{c}_i^{L} = \bar{c}_i^{\text{lc}}, \qquad q_i^{L}(\tau) = \left[\tilde{c}_i^{L}\ ;\ e_{\text{time}}(\tau)\right];$$

- *LS*: local-continuity and self-consistency,

$$\tilde{c}_i^{LS} = \bar{c}_i^{\text{lc}} + \bar{c}_i^{\text{sc}}, \qquad q_i^{LS}(\tau) = \left[\tilde{c}_i^{LS} \;;\; e_{\text{time}}(\tau)\right];$$

- *LSA*: local-continuity, self-consistency, and alignment-consistency,

$$\tilde{c}_i^{LSA} = \bar{c}_i^{\text{lc}} + \bar{c}_i^{\text{sc}} + \bar{c}_i^{\text{ac}}, \qquad q_i^{LSA}(\tau) = \left[\tilde{c}_i^{LSA} \;;\; e_{\text{time}}(\tau)\right].$$

During training, we sample the three paths with probabilities

$$\Pr(L) = 0.1, \qquad \Pr(LS) = 0.1, \qquad \Pr(LSA) = 0.8.$$

At inference time, for each conditioning path, we run the decoder with the corresponding concatenated condition $q_i^L(\tau)$, $q_i^{LS}(\tau)$, or $q_i^{LSA}(\tau)$, convert its endpoint prediction into a tangent velocity estimate using the endpoint-to-velocity construction described earlier, and then combine these tangent velocity estimates linearly. Following the residual decomposition of these three conditioning paths, we perform multi-path residual CFG:

$$v_\tau = v_\tau^L + \lambda_{\text{sc}}\big(v_\tau^{LS} - v_\tau^L\big) + \lambda_{\text{ac}}\big(v_\tau^{LSA} - v_\tau^{LS}\big), \tag{63}$$

where $v_\tau^{LS} - v_\tau^L$ is the self-consistency residual relative to the local-continuity path, and $v_\tau^{LSA} - v_\tau^{LS}$ is the alignment residual relative to the self-consistent path. When $\lambda_{\text{sc}} = \lambda_{\text{ac}} = 1$, the inference reduces to the no-guidance case $v_\tau = v_\tau^{LSA}$, which matches the standard full-condition path.

Although each pathwise velocity is obtained from the endpoint-parameterized direction-only construction described earlier, we do not re-normalize the CFG-combined velocity v_τ back to the fixed magnitude $\pi R/2$. Since all pathwise velocities are tangent vectors at the same bridge state x_τ, their linear combination remains in the same tangent space:

$$v_\tau^L,\, v_\tau^{LS},\, v_\tau^{LSA} \in T_{x_\tau}\mathbb{S}^{D_{\text{tok}}-1}(R) \quad \Longrightarrow \quad v_\tau \in T_{x_\tau}\mathbb{S}^{D_{\text{tok}}-1}(R).$$

Therefore, keeping the CFG-combined magnitude does not violate the spherical geometry. The reason for not fixing the magnitude after CFG is that a fixed velocity norm also fixes the total path length. If one re-normalizes the velocity to $\|v_\tau\|_2 = \pi R/2$ for all $\tau \in [0,1]$, then the generated trajectory has total length

$$\int_0^1 \|v_\tau\|_2 \, d\tau = \frac{\pi R}{2}.$$

This is appropriate when the trajectory is close to the high-dimensional SFM geodesic, whose typical endpoint distance is approximately $\pi R/2$. However, after CFG, the guided vector field may no longer follow the geodesic direction exactly; the resulting trajectory can be curved. A curved trajectory connecting the same conceptual endpoints generally requires a different path length, often larger than the geodesic length. Enforcing a fixed total length in this case can unnecessarily constrain the guided ODE and may prevent the trajectory from reaching the desired endpoint under the learned vector field. By allowing the CFG residuals to change the velocity magnitude, guidance can adjust not only the tangent direction but also the effective integration speed. This gives the guided trajectory additional flexibility while still preserving tangency to the sphere.

Time-dependent CFG guidance. Empirically, we find that extrapolating the self-consistency residual, i.e., using $\lambda_{\text{sc}} > 1$, is important for improving in-context and in-domain consistency, such as speaker-timbre consistency. At the same time, it is also the main source of distribution drift and long-horizon error accumulation. By contrast, extrapolating the alignment residual, i.e., using $\lambda_{\text{ac}} > 1$, is important for strengthening conditional controllability, and we do not observe comparable drift induced by this term. We further hypothesize that the instability associated with self-consistency extrapolation is mainly caused by an unreliable estimate of v_τ^L in the small-τ regime: the L path contains only local continuation information and is therefore substantially less informative than the LS path for estimating the flow output near the early-time bridge regime. As a result, the base point v_τ^L on which the self-consistency extrapolation is applied can be unreliable, and extrapolation from it may produce off-manifold intermediate states whose errors are then amplified through

AR iteration. By contrast, since the alignment residual is defined relative to v_τ^{LS} rather than v_τ^{L}, extrapolation along $v_\tau^{LSA} - v_\tau^{LS}$ is typically stable across the full bridge time range. Motivated by this observation, we use a time-dependent self-consistency guidance weight $\lambda_{\mathrm{sc}}(\tau)$ to avoid extrapolating from an unreliable base in the early-time regime. Concretely, we start from $\lambda_{\mathrm{sc}}(0) = 1$ and gradually increase it to a predefined maximum value $\lambda_{\mathrm{sc}}^{\max}$:

$$\lambda_{\mathrm{sc}}(\tau) = 1 + \big(\lambda_{\mathrm{sc}}^{\max} - 1\big)s(\tau), \qquad s(0) = 0, \quad s(1) = 1, \tag{64}$$

where $s(\tau)$ is a non-decreasing schedule, e.g.,

$$s(\tau) = \tau^{\gamma}, \qquad \gamma > 0.$$

By default, we keep λ_{ac} constant.

Riemannian integration on the sphere. After CFG, the combined velocity remains a tangent vector at the current bridge state:

$$v_\tau \in T_{x_\tau}\mathbb{S}^{D_{\mathrm{tok}}-1}(R).$$

To preserve the spherical constraint during numerical integration, we update the bridge state using the Riemannian exponential map. For a step size $\Delta\tau$, the update is

$$x_{\tau+\Delta\tau} = \mathrm{Exp}_{x_\tau}(\Delta\tau\, v_\tau)\,, \tag{65}$$

where, for any tangent vector $u \in T_x\mathbb{S}^{D_{\mathrm{tok}}-1}(R)$,

$$\mathrm{Exp}_x(u) = \cos\left(\frac{\|u\|_2}{R}\right) x + R\sin\left(\frac{\|u\|_2}{R}\right)\frac{u}{\|u\|_2}. \tag{66}$$

Bridge-time sampling. During training, the bridge time is sampled from a simple mixture distribution. With probability 0.75, we sample

$$a \sim \mathcal{N}(-1, 1), \qquad \tau = \sigma(a) = \frac{1}{1+\exp(-a)},$$

i.e., τ follows a logit-normal distribution biased toward the early, high-noise part of the bridge. With the remaining probability 0.25, we sample

$$\tau \sim \mathrm{Unif}(0, 1).$$

This sampling rule allocates more training probability to small-τ states, where the bridge state is closer to the noisy source and the denoising problem is more ambiguous, while the uniform component guarantees nonzero coverage over the entire bridge interval, including the moderate- and large-τ low-noise regimes.

5 Experiments and Results

In this section, we evaluate Locodec and MP-ELD from two complementary perspectives. First, we study whether the proposed token-space shaping mechanisms preserve reconstruction quality while changing the geometry and statistics of the latent space. Second, we evaluate whether these shaped representations are easier to predict, and whether MP-ELD maintains short-form generation quality while improving long-horizon stability.

5.1 Experimental setup

A central motivation of this work is to examine how much can be gained from representation-space and generation-framework design under moderate model size and restrained budgets relative to large industrial systems. We do not attempt to establish a general scaling law or to claim that design choices universally dominate data scale or model scale. Instead, by using comparatively restrained data and model sizes, and by avoiding external pretrained components and post-training stages, we aim to make the effects of token-space shaping and information routing easier to isolate. This setting allows us to test whether strong reconstruction quality and stable AR generation can be obtained without relying solely on larger datasets, larger models, or additional pretrained inductive biases.

Datasets and evaluation. We train Locodec on an internal bilingual speech dataset of second-scale utterances, and MP-ELD on an internal bilingual dataset consisting of both second-scale and minute-scale utterances, both from real-world recordings. We evaluate both tokenizer reconstruction and generative synthesis on the Seed-TTS-eval [4], where the ZH subset contains 2020 utterances from DiDiSpeech 2 [45] and the EN subset contains 1088 utterances from Common Voice [5]. In addition, we construct a smaller medium-length test set from real-world recordings (ZH only) for efficient CFG grid search, and a long-form test set from real-world recordings (ZH only) to analyze long-horizon AR error accumulation under different CFG configurations.

For tokenizer reconstruction, we report fidelity-oriented metrics, including Mel-cepstral distortion (MCD)[1] [57], STOI[2] [97], and ViSQOL[3] [26]. For both reconstruction and generation, we report task-oriented metrics, including word error rate (WER) and speaker similarity (SIM). The evaluation models and configurations for WER and SIM follow DiTAR [53]. On the long-form test set, we additionally split each generated utterance into non-overlapping 10-second segments and report the segment-level SIM. We also provide spectral visualizations to illustrate the specific acoustic manifestations of AR error accumulation.

Locodec configurations. For 24-kHz stereo audio, Locodec uses a 512-point MDCT window as the signal front-end, and we group 11 adjacent MDCT frames and map them into one token. The resulting frame rate is approximately 8.5 Hz, which we refer to as 8-Hz tokens for simplicity. The native high-dimensional token dimension is fixed to $D_{\text{tok}} = 768$. The encoder consists of 6 FFN blocks followed by a single-step linear downsampling layer. The decoder consists of 6 causal ConvNeXt1D blocks on the token grid and 12 causal ConvNeXt1D blocks on the MDCT coefficient grid. All ConvNeXt1D blocks use kernel size 7 with causal left zero padding, and all FFNs in both the encoder and decoder are SwiGLU FFNs [92]. Under this configuration, the encoder has 59.5M parameters and 5.1G MACs for a one-second input, while the decoder has 180.9M parameters and 12.3G MACs for a one-second input. For higher sampling rates or non-speech audio such as stereo music, the MDCT window size, token rate, and model size can be adjusted accordingly. We leave such extensions outside the scope of this paper.

We evaluate five core-manifold dimensions $d_{\text{core}} \in \{768, 256, 64, 32, 16\}$, where $d_{\text{core}} = 768$ is equivalent to not imposing a lower-dimensional bottleneck. For $d_{\text{core}} \in \{64, 32, 16\}$, we additionally train variants with postfix dimension dropout, abbreviated as PDD hereafter. This gives eight Locodec configurations in total. All tokenizers are trained on 1-second audio clips with a global batch size of 256 seconds for 250k iterations.

MP-ELD configurations. For each Locodec configuration, we train one corresponding MP-ELD model. The MP-ELD architecture contains three token encoders, each consisting of 3 FFN blocks, corresponding to the local-continuity, self-consistency, and alignment-consistency paths. The alignment-consistency LM is a 12-layer RoFormer [95], the self-consistency LM is a 3-layer RoFormer, and the flow-matching decoder is a 3-block FFN network with adaLN-Zero conditioning [85]. All FFNs are also SwiGLU FFNs. The hidden size of the encoders and LMs is 1536, the hidden size of the decoder is 2048, and the dimension of time embedding is 256. The full model has 0.74B parameters. The three token encoders together contain 180.6M parameters and require 1.4G MACs for a one-second (8-token) input. The decoder contains 127.7M parameters and requires 1.0G MACs for a one-second input. All models are trained with a global token budget of 200k tokens (including phonemes and audio embeddings) per iteration for 400k iterations. We maintain an exponential moving average (EMA) version of the model for inference, with the EMA decay set to 0.9995 by default. Inference uses 20 NFEs with Riemannian exponential-map integration on the token sphere.

Following DiTAR, we use phonemes as the text front-end and adopt the same in-context learning (ICL) formulation. During training, each training example is serialized as a phoneme sequence followed by its corresponding audio-token sequence, i.e., (phoneme, audio). The flow-matching next-token prediction loss is applied only to the audio tokens. During inference, the model takes (prompt phoneme, target phoneme, prompt audio) as input and autoregressively continues from the prompt-audio tokens to generates the target-audio tokens.

In practical systems, pretrained components could be introduced at multiple stages of the MP-ELD pipeline.

[1] https://github.com/chenqi008/pymcd
[2] https://github.com/mpariente/pystoi
[3] https://github.com/google/visqol

Table 2 Full-dimensional token reconstruction quality of different Locodec configurations on Seed-TTS-eval. Reconstruction is performed using the native 768-dimensional high-dimensional token.

d_{core}	PDD	ZH					EN				
		MCD ↓	STOI ↑	ViSQOL ↑	WER (%) ↓	SIM ↑	MCD ↓	STOI ↑	ViSQOL ↑	WER (%) ↓	SIM ↑
768	×	2.32	0.98	4.65	1.34	0.740	2.56	0.98	4.65	2.18	0.714
256	×	2.38	0.98	4.64	1.32	0.740	2.64	0.98	4.64	2.16	0.713
64	×	2.27	0.98	4.68	1.33	0.740	2.51	0.98	4.68	2.20	0.715
	✓	2.62	0.97	4.61	1.37	0.736	2.85	0.98	4.63	2.26	0.711
32	×	2.22	0.98	4.68	1.32	0.741	2.44	0.98	4.67	2.15	0.716
	✓	2.60	0.97	4.63	1.36	0.737	2.84	0.98	4.63	2.14	0.712
16	×	2.21	0.98	4.68	1.33	0.742	2.42	0.98	4.68	2.13	0.717
	✓	2.51	0.98	4.63	1.34	0.738	2.72	0.98	4.64	2.19	0.713

For example, the alignment-consistency path could use a pretrained text LM or a pretrained cross-modal alignment model, while the token encoders or condition encoders could be initialized from strong SSL or ASR models to provide additional semantic or acoustic inductive biases. These extensions are compatible with the MP-ELD formulation and may further improve performance. In this work, however, we deliberately avoid using pretrained models in any MP-ELD component and train all encoders, LMs, and decoders from scratch. This prevents external pretrained representations from dominating the information layout and allows us to test whether the proposed local-continuity, self-consistency, and alignment-consistency routing behavior can emerge from the training objective and architecture alone.

5.2 Results on reconstruction

We first evaluate whether the proposed token-space shaping mechanisms affect tokenizer reconstruction quality. Table 2 reports full-dimensional token reconstruction quality on Seed-TTS-eval, where reconstruction is performed using the native 768-dimensional high-dimensional token. Under our training configuration, imposing a low-dimensional core manifold does not degrade full-dimensional token reconstruction quality. Across $d_{\text{core}} \in \{768, 256, 64, 32, 16\}$ without PDD, all metrics remain nearly unchanged. In fact, the small variations across core dimensions are comparable to normal training variation, and no consistent loss of reconstruction fidelity is observed as the core dimension is reduced. This indicates that the low-dimensional reconstruction path can reshape the geometry of the high-dimensional token space without noticeably reducing the information preserved by the full native token.

PDD has a slightly different effect. Compared with the corresponding non-PDD configurations, PDD leads to a mild degradation in MCD, suggesting a small increase in spectral mismatch. However, its effect on the remaining metrics is minimal: STOI and ViSQOL remain almost unchanged, and the task-oriented WER and SIM scores show only negligible differences. As all metrics except MCD are computed at 16 kHz, the fact that PDD mainly affects MCD while leaving STOI, ViSQOL, WER, and SIM essentially unchanged suggests that the induced coordinate-wise energy hierarchy primarily introduces a mild mismatch in less critical spectral details, especially in the middle- and high-frequency regions, while preserving the core low- and mid-frequency information needed for intelligibility, perceptual quality, and speaker identity.

To provide an intuitive view of the energy bias induced by PDD, Figure 3 visualizes the average coordinate-wise token energy. Without PDD, the native high-dimensional token energy is nearly uniform across dimensions. With PDD, prefix dimensions acquire substantially larger energy, postfix dimensions acquire substantially smaller energy, and the logarithm of the per-dimension energy decays approximately linearly with the dimension index. This indicates that the combination of noise injection and PDD effectively induces a stable coordinate-wise energy bias through training dynamics, without explicitly prescribing a target energy profile.

We further examine the relationship between the explicitly learned low-dimensional core and the prefix subspace induced by PDD. This analysis is meaningful only for PDD-enabled tokenizers, because without PDD the coordinate order is not trained to carry a prefix-to-postfix availability hierarchy. We therefore evaluate the three PDD-enabled configurations with $d_{\text{core}} \in \{16, 32, 64\}$. For each configuration, we compare reconstruction

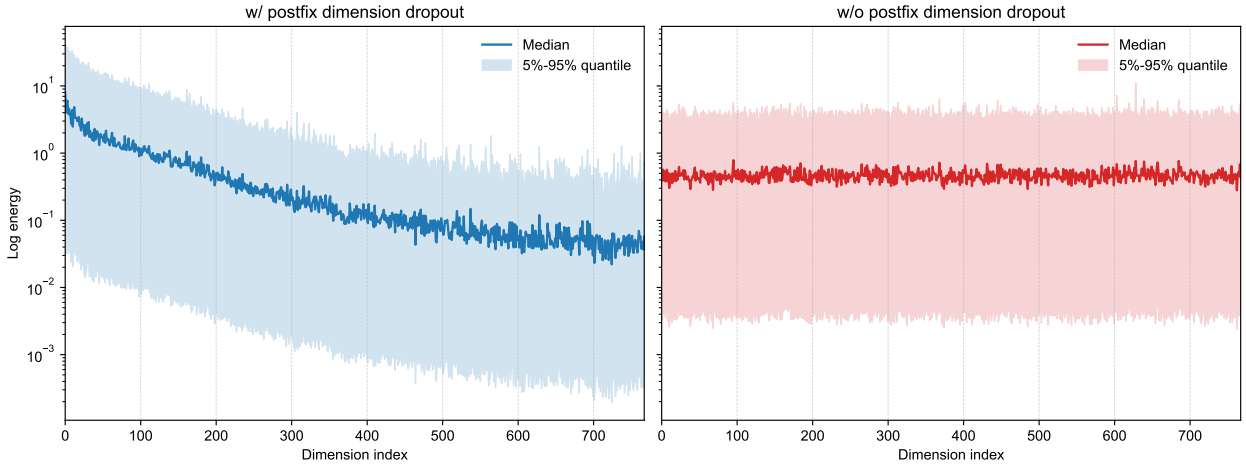

Figure 3 Coordinate-wise energy profiles of Locodec tokens. When combined with rotation noise injection, PDD automatically induces a stable prefix-to-postfix energy hierarchy, while models without postfix dropout keep an almost uniform energy distribution across dimensions.

Table 3 Reconstruction from restricted representations for PDD-enabled Locodec models. "Core" denotes reconstruction from the lifted low-dimensional token. "Prefix-K" denotes reconstruction after keeping only the first K dimensions of the native high-dimensional token and zeroing out the remaining dimensions.

d_{core}	Repr.	ZH					EN				
		MCD ↓	STOI ↑	ViSQOL ↑	WER (%) ↓	SIM ↑	MCD ↓	STOI ↑	ViSQOL ↑	WER (%) ↓	SIM ↑
16	Core	6.53	0.83	3.15	15.81	0.396	6.63	0.83	3.29	23.00	0.284
	Prefix-16	7.93	0.75	2.78	62.90	0.334	8.09	0.76	2.86	72.37	0.164
	Prefix-32	6.45	0.84	3.23	18.63	0.411	6.62	0.84	3.36	24.65	0.295
	Prefix-64	5.11	0.90	3.71	3.86	0.540	5.35	0.90	3.83	6.46	0.482
	Full	2.51	0.98	4.63	1.34	0.738	2.72	0.98	4.64	2.19	0.713
32	Core	5.24	0.89	3.60	4.92	0.484	5.47	0.89	3.74	7.69	0.417
	Prefix-16	8.03	0.74	2.72	60.58	0.335	8.17	0.75	2.85	72.11	0.160
	Prefix-32	6.35	0.83	3.26	18.30	0.426	6.47	0.84	3.41	24.42	0.304
	Prefix-64	5.09	0.89	3.70	4.14	0.540	5.30	0.90	3.82	6.21	0.477
	Full	2.60	0.97	4.63	1.36	0.737	2.84	0.98	4.63	2.14	0.712
64	Core	4.50	0.92	3.96	2.47	0.595	4.77	0.92	4.06	3.87	0.553
	Prefix-16	7.93	0.74	2.77	61.94	0.344	7.96	0.75	2.91	68.41	0.187
	Prefix-32	6.31	0.83	3.27	16.92	0.426	6.44	0.84	3.42	26.25	0.306
	Prefix-64	5.00	0.90	3.73	4.10	0.548	5.21	0.90	3.85	5.98	0.483
	Full	2.62	0.97	4.61	1.37	0.736	2.85	0.98	4.63	2.26	0.711

from three restricted representations: the lifted low-dimensional core token, the prefix-K subspace of the native high-dimensional token, and the full native token. For prefix-K reconstruction, we keep only the first K coordinates of the native token and set all remaining coordinates to zero before feeding the token to the decoder, without renormalization. This matches the form of the training-time postfix dropout corruption.

The results are reported in Table 3. They show that the prefix subspaces indeed form a functional hierarchy: within each PDD-enabled tokenizer, reconstruction quality improves consistently from Prefix-16 to Prefix-32 and then to Prefix-64. This confirms that the energy profile induced by PDD is not merely a statistical artifact, but corresponds to an actual ordering of decodable information, in a manner loosely analogous to the coarse-to-fine information hierarchy in RVQ. At the same time, the core dimension is not equivalent to the prefix dimension: for the same nominal dimensionality, Core-d consistently outperforms Prefix-d, even though the low-dimensional reconstruction path is down-weighted during training. This indicates that the

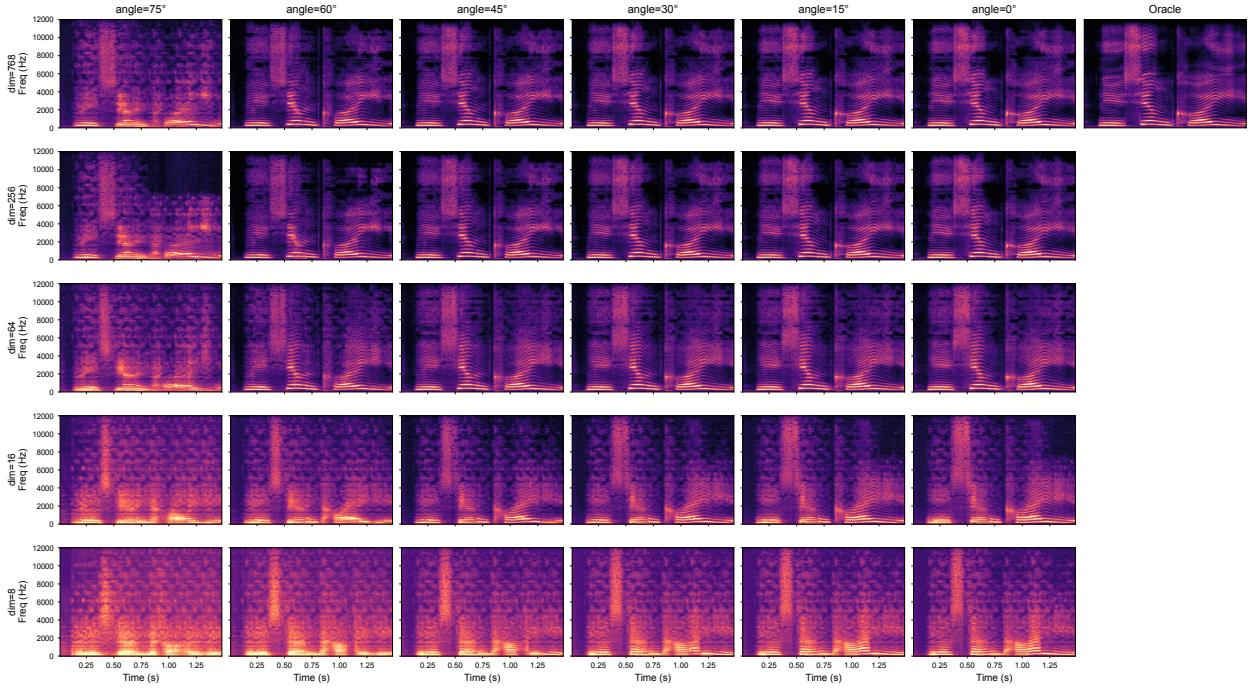

Figure 4 Spectral visualization of Locodec reconstructions under angular token corruption and prefix restriction. Columns correspond to different rotation angles and rows correspond to different retained prefix dimensions.

low-dimensional path maintains an effective shaping pressure on the token space. We also observe that the reconstruction quality of a fixed prefix length is largely stable across different choices of d_{core}. For example, Prefix-64 gives very similar reconstruction metrics for $d_{\text{core}} = 16, 32$, and 64. This suggests that the native coordinate hierarchy is mainly determined by the PDD availability bias, whereas the choice of d_{core} primarily affects the quality of the explicit core representation. Finally, Prefix-K is not constrained to have a fixed norm within its active K-dimensional subspace: after masking, its norm varies and its coordinate energies are highly non-uniform. In this sense, it is geometrically less constrained than the fixed-radius core representation and in principle has an additional radial degree of freedom. Nevertheless, Prefix-K remains weaker than Core-K at the same nominal dimension, including at $K = 64$. A different dropout schedule or substantially longer training might improve fixed-prefix reconstruction, but we leave this direction outside the scope of the present study. Taken together, these results suggest that the low-dimensional constraint and PDD play complementary roles, and that they do not introduce severe conflicts that would compromise full-dimensional token reconstruction quality.

As discussed in Section 3, PDD is not used as an isolated masking trick: its availability bias is combined with spherical noise injection, so that dimensions that remain available more frequently are also encouraged to become more identifiable under angular corruption. To provide an intuitive view of the reconstruction behavior induced by this joint training scheme, Figure 4 visualizes reconstructions from the PDD-enabled tokenizer with $d_{\text{core}} = 16$ under different rotation angles and retained prefix dimensions. We rotate the high-dimensional token by angles $\theta \in \{0°, 15°, 30°, 45°, 60°, 75°\}$, and decode from prefix dimensions $K \in \{8, 16, 64, 256, 768\}$. The visualization shows how reconstruction quality changes as angular corruption becomes stronger and as fewer prefix dimensions are retained, providing a qualitative view of both decoder robustness and the PDD-induced information hierarchy. Notably, the full-dimensional token still preserves reasonably good spectral quality even under large angular perturbations, such as 60° in the visualization. This behavior is encouraged by the training-time spherical corruption, whose maximum rotation angle is 90°, and suggests that the decoder learns a relatively large reconstruction-stable basin around each token. However, as discussed earlier, learning such large basins in an unconstrained high-dimensional space can encourage different tokens to become widely separated, or even nearly orthogonal, thereby weakening interpolatability. The low-dimensional

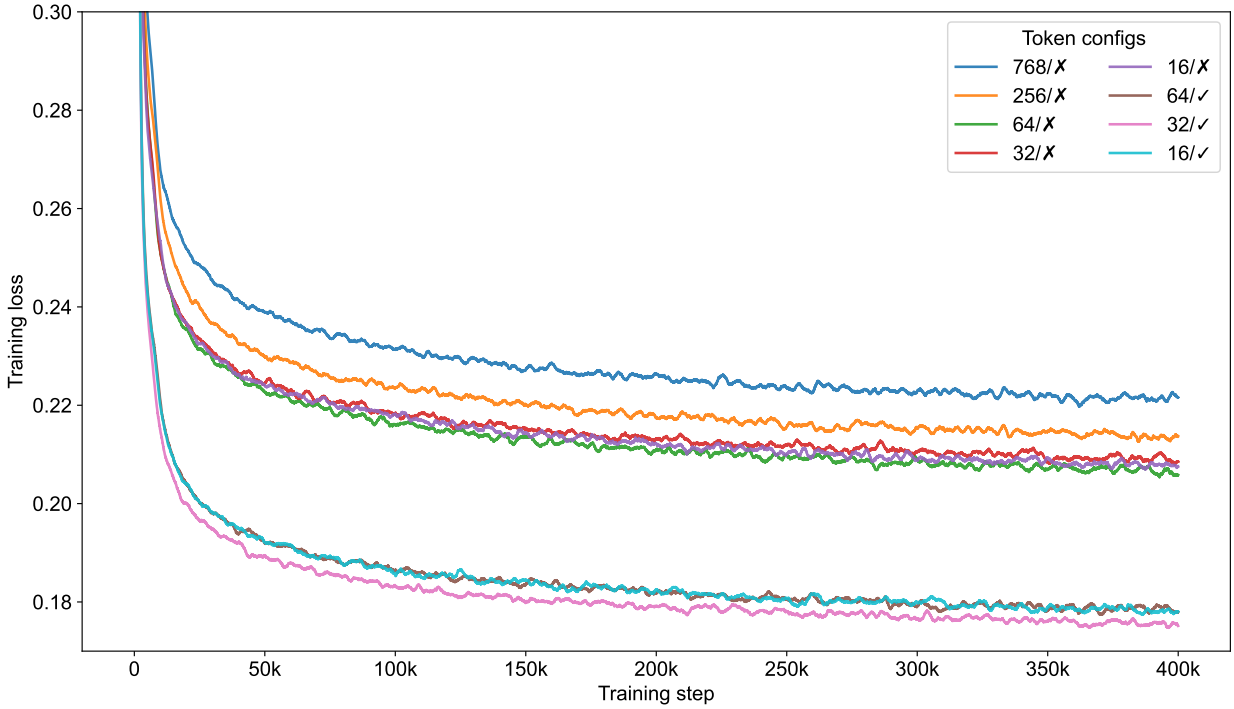

Figure 5 Training curves of the MP-ELD cosine direction loss for different Locodec configurations. The notation d/PDD denotes the core dimension and whether PDD is used. All MP-ELD models use the same architecture and training setup. Faster convergence and lower final loss indicate that the corresponding token representation is easier to predict under the fixed generator configuration.

core constraint is therefore important for preventing robustness from relying purely on high-dimensional angular separation. We return to this point in the generation results below.

5.3 Results on generation

We next evaluate how different Locodec representations affect AR continuous-token generation. For each of the eight tokenizer configurations, we train an MP-ELD model with the same architecture, training data, optimization setup, and training budget. Under this controlled setting, the convergence behavior and absolute value of the generative-model training loss provide an empirical proxy for token predictability: if the same generator converges faster and reaches a lower training loss on one token representation than on another, then the former representation can be regarded as easier to model under the fixed generator configuration.

Figure 5 shows the MP-ELD cosine direction loss curves for the eight Locodec configurations. The first clear trend is that imposing a low-dimensional core manifold substantially improves predictability. The no-bottleneck configuration $768/\times$ converges the slowest and reaches the highest final loss. Reducing the core dimension to 256 already lowers the loss noticeably, and further reducing it to 64, 32, or 16 gives an additional improvement. However, this improvement is not monotonic without limit. Once the core dimension becomes sufficiently small, the gain largely saturates: the non-PDD configurations with $d_{\text{core}} \in \{64, 32, 16\}$ converge to similar losses. This suggests that the low-dimensional constraint helps organize the high-dimensional token space into a more predictable geometry, but making the core dimension arbitrarily small does not continue to provide proportional benefits.

The second and more pronounced trend is the effect of PDD. For all core dimensions, PDD-enabled models converge faster and reach substantially lower losses than their non-PDD counterparts. This indicates that the coordinate-wise energy bias induced by PDD provides an additional and strong improvement in single-token predictability beyond the low-dimensional manifold constraint. Among the PDD-enabled configurations,

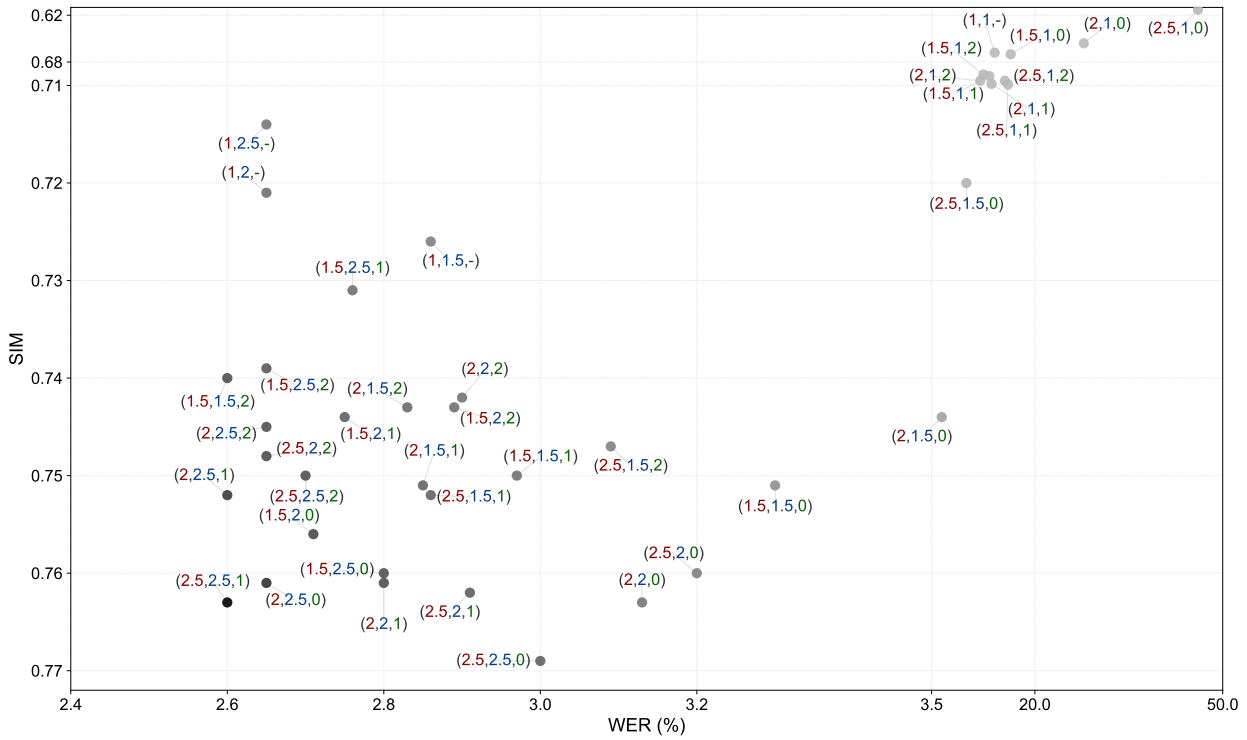

Figure 6 WER/SIM trade-off under different CFG configurations on the internal CFG-selection set. Each point corresponds to one setting from the sweep over $(\lambda_{\mathrm{sc}}^{\max}, \lambda_{\mathrm{ac}}, \gamma)$. The scatter shows that alignment-consistency guidance mainly affects WER, while self-consistency guidance mainly affects SIM, and that the bridge-time schedule controls the trade-off induced by self-consistency extrapolation. Darker points indicate configurations closer to the preferred low-WER, high-SIM region.

$d_{\mathrm{core}} = 32$ achieves the lowest final loss, while $d_{\mathrm{core}} = 16$ and 64 are slightly worse. Thus, from the perspective of generator-side predictability, the non-PDD and PDD comparisons together provide empirical support for our assumption that a moderate low-dimensional core manifold can make a high-dimensional token space easier to model, and that the additional identifiability induced by PDD can further improve this predictability.

Since the $d_{\mathrm{core}} = 32$, PDD-enabled tokenizer achieves the lowest training loss among all tested configurations, we use it as the default representation to study the effect of CFG configurations on the balance between WER and SIM. We sweep CFG hyperparameters on a smaller internal evaluation set over a predefined grid:

$$\lambda_{\mathrm{sc}}^{\max} \in \{1, 1.5, 2, 2.5\}, \qquad \lambda_{\mathrm{ac}} \in \{1, 1.5, 2, 2.5\}, \qquad \gamma \in \{0, 1, 2\}.$$

Here $\gamma = 0$ denotes constant self-consistency guidance, i.e., $s(\tau) \equiv 1$, while $\gamma > 0$ uses the delayed schedule $s(\tau) = \tau^{\gamma}$. The setting $\lambda_{\mathrm{sc}}^{\max} = \lambda_{\mathrm{ac}} = 1$ corresponds to the standard full-condition path $v_\tau = v_\tau^{LSA}$ without CFG. When $\lambda_{\mathrm{sc}}^{\max} = 1$, the choice of γ has no effect. This gives a total of 40 configurations.

Figure 6 shows the resulting WER/SIM trade-off. The first clear trend is that the alignment-consistency guidance scale is the main factor controlling whether the model enters a content-aligned generation regime. When $\lambda_{\mathrm{sc}}^{\max} = 1$, increasing λ_{ac} from 1 to 1.5, 2, and 2.5 moves the model from a high-WER region to a much lower-WER region, with WER decreasing from above 10% to roughly the 2.6–2.9% range. In this process, SIM first improves and then degrades, indicating that stronger alignment guidance mainly improves content following, but excessive alignment extrapolation may start to trade off against acoustic similarity. This behavior is consistent with the intended role of the alignment-consistency residual $v_\tau^{LSA} - v_\tau^{LS}$: it primarily acts as an external-condition alignment correction rather than a general acoustic-consistency correction.

The effect of self-consistency guidance is different. In the valid low-WER region, increasing $\lambda_{\mathrm{sc}}^{\max}$ tends

to improve SIM more directly than WER. For example, at comparable alignment strength, increasing the self-consistency scale from 1.5 to 2 or 2.5 can move SIM from the mid-0.74 range to around 0.76, while WER changes more mildly and remains within the same general low-WER regime. This supports the interpretation that the self-consistency residual $v_\tau^{LS} - v_\tau^{L}$ mainly strengthens modality-internal acoustic consistency, including speaker identity and slowly varying acoustic state.

However, stronger self-consistency guidance does not imply a monotonic SIM improvement. When the alignment-consistency scale is insufficient, such as $\lambda_{\text{ac}} = 1$, increasing $\lambda_{\text{sc}}^{\max}$ does not rescue the high WER. Moreover, SIM also becomes non-monotonic and can drop substantially under some configurations. This suggests that if the model is not properly aligned to the external condition, amplifying self-consistency residuals may instead reinforce an unstable acoustic trajectory. In such cases, accumulated acoustic distortion can degrade not only intelligibility but also the speaker-similarity metric itself.

The schedule parameter γ further controls how aggressively self-consistency guidance is applied over the bridge time. For the same $(\lambda_{\text{sc}}^{\max}, \lambda_{\text{ac}})$, constant guidance with $\gamma = 0$ often gives stronger SIM, but can also move the point away from the best WER region. For example, under strong guidance scales such as $(\lambda_{\text{sc}}^{\max}, \lambda_{\text{ac}}) = (2.5, 2.5)$, using $\gamma = 0$ yields very high SIM but a higher WER, while delayed schedules such as $\gamma = 1$ or $\gamma = 2$ reduce early-time self-consistency extrapolation and give a more balanced WER/SIM trade-off. This is consistent with the motivation of the time-dependent guidance schedule: self-consistency guidance is useful, but applying it too strongly in the early high-noise part of the bridge can make the residual extrapolation less reliable.

Taken together, the sweep provides empirical evidence that MP-ELD learns a meaningful degree of information routing from scratch. Nevertheless, changing λ_{ac} and $\lambda_{\text{sc}}^{\max}$ produces different and interpretable movements in the WER/SIM plane: the alignment-consistency residual mainly affects content alignment, while the self-consistency residual mainly affects acoustic and speaker consistency. The grid search also identifies a set of Pareto-competitive CFG configurations, from which we select the best overall WER/SIM trade-off for comparison with prior systems.

To better understand what these WER/SIM differences correspond to acoustically, we visualize one representative utterance generated with different CFG configurations in Figure 7. All six spectrograms are generated from the same input sample using the same model and tokenizer described above, differing only in the CFG configuration. The examples are ordered from top-left to bottom-right by increasing SIM for this utterance: the first row shows the two configurations with the lowest SIM, the second row shows two middle-SIM configurations, and the third row shows the two configurations with the highest SIM. Compared with the WER/SIM scatter plot alone, this visualization gives a more direct view of how different guidance choices affect the acoustic structure of the generated signal.

We can first observe that severe AR error accumulation clearly affects the metrics: when strong drift or collapse is visible, at least one of WER and SIM becomes poor. At the same time, the onset time of error accumulation is not fixed: in the first row, degradation appears very early and dominates most of the generated waveform, while in other configurations such as the middle-right and bottom-left examples, the generation is initially reasonable but begins to drift after several seconds. The acoustic manifestation of drift is also not unique: we observe elongated or locally repeated harmonic structures, gradually increasing noise-like energy, and frequency bands whose energy slowly increases or decreases over time. Note that other failure modes, such as gradually increasing speaking rate or global loudness drift, are also observed in practice but are omitted here for space.

The first row illustrates the importance of sufficient alignment-consistency guidance. Both examples use $\lambda_{\text{ac}} = 1$ and $\gamma = 0$, and both show severe error accumulation. The no-extrapolation configuration $(1, 1, 0)$ already fails to maintain a valid content-aligned trajectory on this sample, leading to WER = 100%. Increasing self-consistency guidance alone, as in $(2.5, 1, 0)$, does not fix this failure; instead, the generation collapses into a long, acoustically self-reinforcing trajectory with even lower SIM. This supports the interpretation from the CFG sweep that the self-consistency residual cannot replace the alignment-consistency residual. Without sufficient alignment guidance, amplifying self-consistency may reinforce an incorrect acoustic continuation rather than recover the intended content.

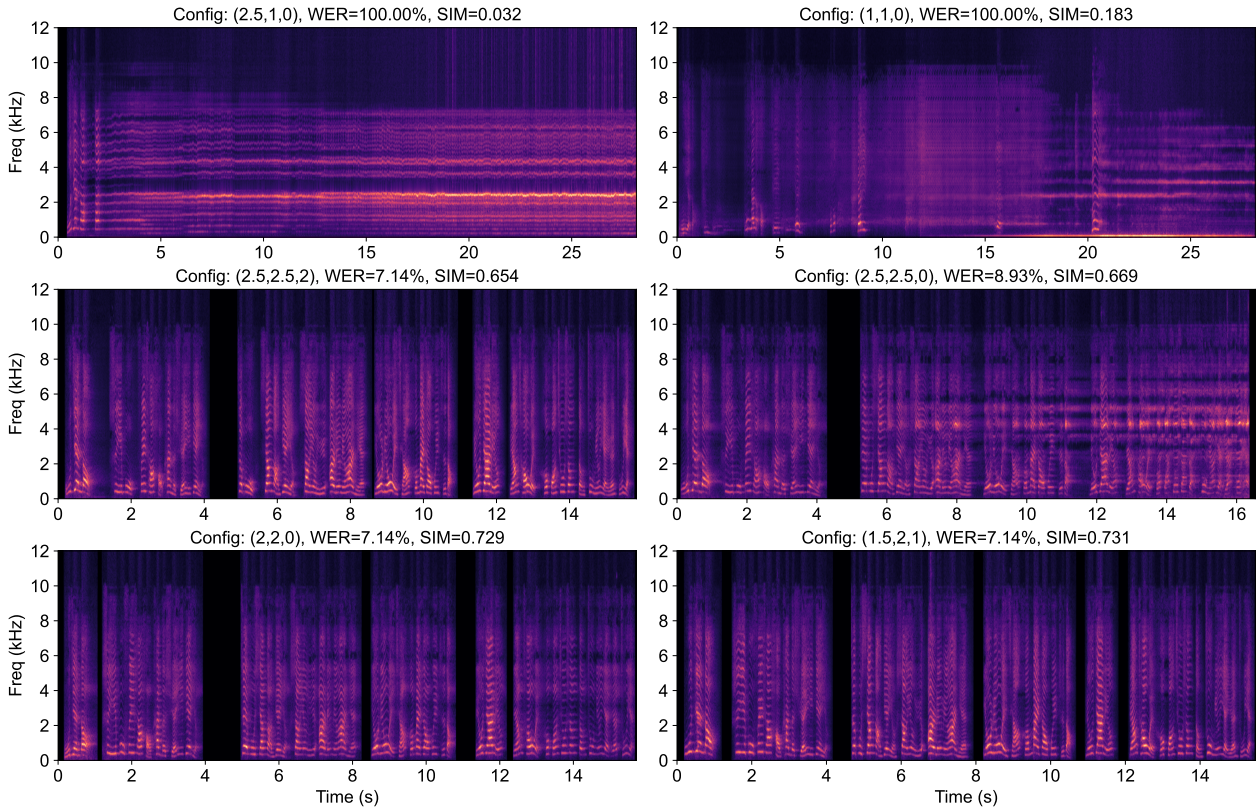

Figure 7 Spectrograms of the same utterance generated under different CFG configurations. Each title reports $(\lambda_{\text{sc}}^{\max}, \lambda_{\text{ac}}, \gamma)$, WER, and SIM for the generated sample. The six examples are ordered from top-left to bottom-right by increasing SIM. The visualization shows that different CFG configurations lead to different forms of AR error accumulation, including early collapse, delayed spectral drift, harmonic repetition, and frequency-band energy drift.

The second row isolates the effect of the time schedule γ. The two configurations have the same guidance scales, $(\lambda_{\text{sc}}^{\max}, \lambda_{\text{ac}}) = (2.5, 2.5)$, and differ only in the self-consistency schedule. With constant self-consistency guidance, $\gamma = 0$, the spectrogram starts to show visible drift around 7 seconds: energy around the 4–5 kHz region gradually increases, and neighboring frequency bands also exhibit different degrees of energy amplification or attenuation. With delayed guidance, $\gamma = 2$, the spectrum remains more stable over the utterance. Interestingly, the $\gamma = 0$ configuration has a slightly higher utterance-level SIM despite having a higher WER and more visible spectral distortion. This indicates that utterance-level SIM alone is not sufficient to judge the actual acoustic quality or stability of generated speech.

The third row further shows that even similar WER and SIM values do not necessarily imply identical acoustic behavior. The two configurations obtain almost the same utterance-level metrics, but the bottom-left example still shows mild frequency-band drift after around 7 seconds, especially near the 3 kHz and 6 kHz regions, whereas the bottom-right example remains more stable throughout the utterance. This suggests that automatic metrics such as WER and SIM are useful but incomplete summaries of generation quality, as they may fail to capture subtle spectral drift or slowly accumulating acoustic artifacts even when the final utterance-level scores are similar.

To further examine this limitation, we evaluate long-form generation using both global-level and segment-level metrics on the long-form evaluation set. Based on the CFG sweep above, we select 18 representative configurations from the core operating region,

$$\lambda_{\text{sc}}^{\max} \in \{2, 2.5\}, \qquad \lambda_{\text{ac}} \in \{1.5, 2, 2.5\}, \qquad \gamma \in \{0, 1, 2\}.$$

For each generated utterance, we keep the first 50 seconds, split it into five non-overlapping 10-second segments,

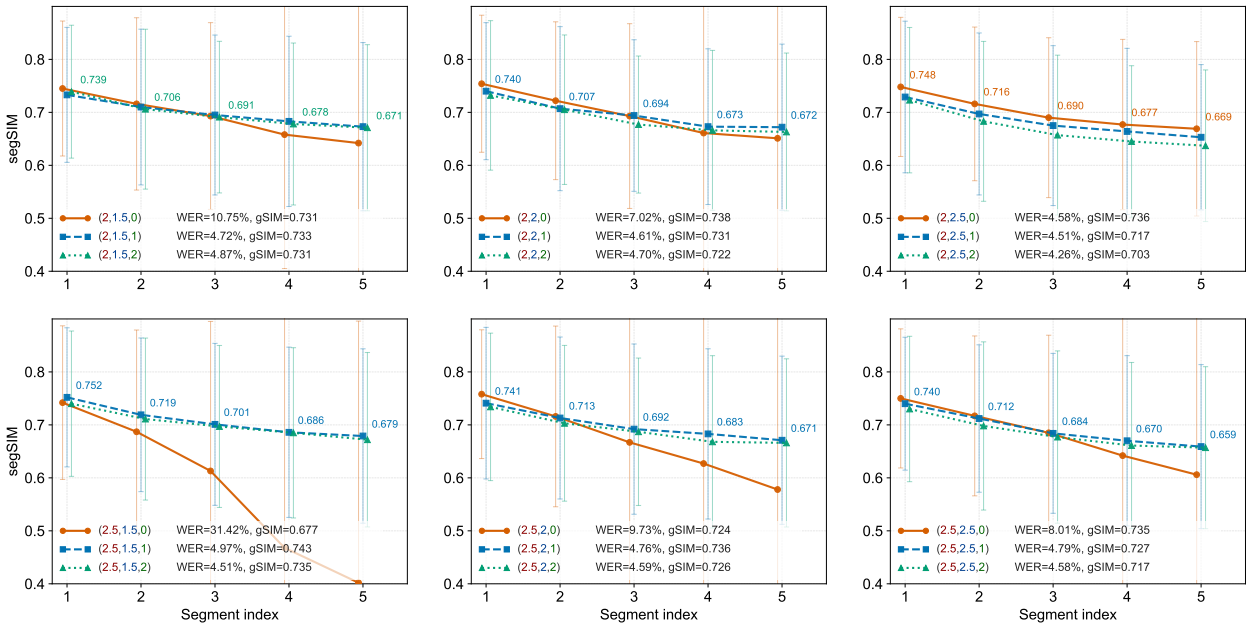

Figure 8 Long-form segment-level SIM under representative CFG configurations. Each generated utterance is split into five non-overlapping 10-second segments. Each subplot fixes $(\lambda_{\text{sc}}^{\max}, \lambda_{\text{ac}})$ and compares different γ values. Curves show dataset-level mean segment SIM, and error bars indicate 95% confidence intervals. The legend reports global WER and global SIM. The annotated values correspond to the configuration with the best overall stability–quality trade-off in each subplot.

and compute SIM for each segment. Figure 8 reports the dataset-level mean segment SIM together with 95% confidence intervals. Each subplot fixes $(\lambda_{\text{sc}}^{\max}, \lambda_{\text{ac}})$ and compares different γ values. We also report the global WER and global SIM (gSIM) in the legend. For readability, in each subplot we annotate the segment-wise mean SIM values of the configuration with the best overall stability–quality trade-off.

The first observation is that constant self-consistency guidance, i.e., $\gamma = 0$, is substantially less stable in long-form generation, especially when $\lambda_{\text{sc}}^{\max}$ is large. For example, with $(\lambda_{\text{sc}}^{\max}, \lambda_{\text{ac}}, \gamma) = (2.5, 1.5, 0)$, the segment SIM drops from 0.742 in the first segment to 0.402 in the last segment, and the global WER increases to 31.42%. Similar but less extreme degradation is also observed for $(2.5, 2, 0)$ and $(2.5, 2.5, 0)$. In contrast, delayed schedules with $\gamma > 0$ greatly reduce this long-horizon decay. For the same $(\lambda_{\text{sc}}^{\max}, \lambda_{\text{ac}}) = (2.5, 1.5)$, changing γ from 0 to 1 improves the last-segment SIM from 0.402 to 0.679, while reducing WER from 31.42% to 4.97%. This suggests that the bridge-time schedule is a key factor in preventing self-consistency guidance from becoming a source of accumulated drift. At the same time, while a larger γ typically improves the long-horizon stability and especially the spectrogram quality, it also delays self-consistency extrapolation more aggressively and may reduce the overall speaker-similarity level. In four of the six subplots, the best stability–quality trade-off is obtained with $\gamma = 1$, which provides a middle ground: it avoids strong self-consistency extrapolation in the early high-noise regime while still applying sufficient self-consistency guidance later in the bridge. This is consistent with the medium-length spectrogram examples above, where delayed guidance reduces spectral drift without completely suppressing the benefit of self-consistency guidance.

The curves further illustrate the different roles of $\lambda_{\text{sc}}^{\max}$ and λ_{ac}. Stronger self-consistency guidance can produce higher similarity near the beginning of an utterance, but it may also increase the risk of later drift. For example, $(2.5, 2, 0)$ achieves a very high first-segment SIM of 0.758, but its last-segment SIM drops to 0.578. By contrast, a more balanced scheduled configuration such as $(2, 2, 1)$ starts from a slightly lower first-segment SIM of 0.740, but decays more slowly and remains at 0.672 in the last segment. Thus, the highest initial similarity is not necessarily a good indicator of long-form stability. This also shows why global SIM alone can be misleading: two configurations may have very similar utterance-level SIM but very different temporal behavior. For instance, $(2.5, 2.5, 0)$ and $(2.5, 2, 1)$ have nearly identical global SIM values, 0.735

Table 4 Generation performance of different Locodec token configurations under three representative CFG settings. CFG-L denotes the most stable long-form configuration, $(\lambda_{\text{sc}}^{\max}, \lambda_{\text{ac}}, \gamma) = (2, 2, 1)$. CFG-M denotes the best overall configuration selected on the medium-length CFG search set, $(\lambda_{\text{sc}}^{\max}, \lambda_{\text{ac}}, \gamma) = (2.5, 2.5, 1)$. CFG-S denotes the configuration with the highest first-segment (short-horizon) SIM on the long-form set, $(\lambda_{\text{sc}}^{\max}, \lambda_{\text{ac}}, \gamma) = (2.5, 2, 0)$.

Token	CFG	Seed-TTS-eval				Long-form set						
		ZH		EN		WER (%) ↓	gSIM ↑	SegSIM				
		WER (%) ↓	SIM ↑	WER (%) ↓	SIM ↑			Seg1 ↑	Seg2 ↑	Seg3 ↑	Seg4 ↑	Seg5 ↑
768/×	L	1.31	0.683	1.90	0.622	7.82	0.716	0.732	0.694	0.670	0.644	0.639
	M	1.12	0.683	1.85	0.624	8.36	0.710	0.731	0.696	0.669	0.628	0.624
	S	2.04	0.691	2.15	0.625	25.35	0.687	0.743	0.677	0.623	0.522	0.448
256/×	L	1.11	0.682	1.86	0.614	5.81	0.729	0.737	0.705	0.687	0.675	**0.680**
	M	1.04	0.686	1.88	0.618	4.84	0.725	0.742	0.701	0.687	**0.677**	0.667
	S	1.46	0.692	5.23	0.620	15.94	0.714	0.747	0.696	0.643	0.568	0.512
64/×	L	1.09	0.683	1.94	0.620	4.91	0.717	0.726	0.702	0.688	0.669	0.660
	M	1.50	0.686	1.82	0.627	4.69	0.716	0.728	0.697	0.676	0.662	0.666
	S	1.56	0.691	2.26	0.625	9.68	0.713	0.738	0.690	0.654	0.611	0.566
32/×	L	1.07	0.675	2.08	0.608	4.64	0.708	0.721	0.683	0.671	0.656	0.642
	M	1.12	0.679	2.16	0.615	5.44	0.705	0.723	0.688	0.656	0.630	0.621
	S	1.52	0.683	2.47	0.615	20.80	0.674	0.734	0.664	0.583	0.487	0.419
16/×	L	1.25	0.672	1.95	0.603	6.23	0.714	0.726	0.696	0.680	0.648	0.646
	M	1.30	0.678	1.93	0.609	9.49	0.702	0.729	0.686	0.658	0.633	0.594
	S	2.58	0.677	2.98	0.607	34.77	0.630	0.740	0.661	0.510	0.413	0.273
64/✓	L	1.11	0.680	2.00	0.616	4.96	0.726	0.734	0.705	0.690	0.673	0.668
	M	1.12	0.683	1.87	0.620	4.78	0.720	0.729	0.697	0.684	0.667	0.663
	S	1.54	0.695	2.05	**0.630**	9.63	**0.748**	**0.762**	**0.720**	0.683	0.628	0.574
32/✓	L	**0.95**	0.687	1.87	0.615	4.61	0.731	0.740	0.707	**0.694**	0.673	0.672
	M	0.99	0.691	1.80	0.622	4.79	0.727	0.740	0.712	0.684	0.670	0.659
	S	1.24	0.697	2.09	0.628	9.73	0.724	0.758	0.716	0.667	0.627	0.578
16/✓	L	1.66	0.688	1.94	0.619	4.80	0.726	0.732	0.700	0.686	0.674	0.676
	M	1.29	0.691	**1.75**	0.624	**4.55**	0.718	0.734	0.702	0.684	0.666	0.670
	S	1.81	**0.698**	2.47	**0.630**	10.18	0.725	0.746	0.708	0.680	0.649	0.601

and 0.736, respectively. However, their last-segment SIM values differ substantially: 0.606 for $(2.5, 2.5, 0)$ versus 0.671 for $(2.5, 2, 1)$. The former therefore hides a much stronger long-horizon degradation behind a similar global score. Moreover, although Figure 8 reports only WER and segment-level SIM, our qualitative inspection shows phenomena consistent with Figure 7: even configurations with similar segment-level SIM can exhibit noticeably different spectral quality. This further indicates that long-form audio generation should not be evaluated solely by a small set of automatic metrics, and a more complete evaluation should combine signal-level analysis, perceptual listening, and task-oriented automatic metrics. Empirically, we find that configurations with $\lambda_{\text{ac}} \geq \lambda_{\text{sc}}^{\max}$ tend to be safer for long-form generation, especially in terms of spectral and temporal stability, although they often sacrifice some global or segment-level SIM. Conversely, configurations with $\lambda_{\text{sc}}^{\max} > \lambda_{\text{ac}}$ can achieve stronger short-range or early-segment speaker similarity, but are more sensitive to the guidance schedule and more prone to accumulated drift. This suggests that the optimal CFG operating point may depend on the target generation length: short utterances may prefer stronger self-consistency guidance, whereas long-form generation may require more conservative or more alignment-dominant guidance. A natural extension is to use dynamic CFG schedules that vary not only with bridge time τ, but also with AR generation time. For example, one may gradually adjust the relative strength of self-consistency and alignment-consistency guidance as generation proceeds. We leave this direction for future work.

The preceding CFG analyses are based on the 32/✓ tokenizer configuration. We now extend the comparison to the other seven token configurations and examine whether the above observations remain consistent.

Table 5 Generation results on Seed-TTS-eval. For MP-ELD, we report the result under its best CFG configuration from Table 4. All benchmark systems are AR models, and we select the version without post-training when applicable. The "Tok./LM rate" column denotes the native token frame rate and the effective frame rate seen by the LM, respectively.

Model	Tok./LM rate (Hz)	ZH		EN	
		WER (%) ↓	SIM ↑	WER (%) ↓	SIM ↑
Human	—	1.25	0.755	2.14	0.734
CosyVoice 2 [37]	25/25	1.45	0.748	2.57	0.652
CosyVoice 3-1.5B [38]	25/25	1.12	0.781	2.21	0.720
FireRedTTS [44]	25/25	1.51	0.635	3.82	0.460
FireRedTTS-2 [110]	12.5/12.5	1.14	0.736	1.95	0.665
Spark TTS [104]	50/50	1.20	0.672	1.98	0.584
VibeVoice [86]	7.5/7.5	1.16	0.744	3.04	0.689
DiTAR [53]	40/10	1.02	0.753	1.69	0.735
VoxCPM2 [127]	25/6.25	0.97	0.795	1.84	0.753
dots. tts [65]	25/6.25	0.96	**0.805**	**1.34**	**0.768**
MP-ELD, 64/✓	8/8	1.54	0.695	2.05	0.630
MP-ELD, 32/✓	8/8	**0.95**	0.687	1.87	0.615
MP-ELD, 16/✓	8/8	1.29	0.691	1.75	0.624

Table 4 compares the eight Locodec token configurations under three representative CFG settings: CFG-L is the relatively most stable long-form configuration, CFG-M is the best overall configuration selected on the medium-length CFG search set, and CFG-S is the configuration that favors short-horizon SIM.

Several observations follow. First, the high-dimensional configurations without a very low-dimensional core are still learnable. In particular, the 768/× and 256/× tokenizers already obtain competitive short-form WER on Seed-TTS-eval, and the 256/× configuration remains reasonably stable on the long-form set under CFG-L and CFG-M. This suggests that the Locodec design itself, including spherical normalization, strong angular corruption, and decoder robustness training, already imposes useful structure on the high-dimensional token space. The low-dimensional core and PDD further improve predictability and stability, but they are not the only source of learnability.

Second, all token configurations achieve relatively strong WER on Seed-TTS-eval, and this may be partly explained by the native low frame rate of Locodec. At 8 Hz, each token covers roughly 125 ms of audio, which is close to the duration scale of phoneme-level content. Thus, it seems that a low-frame-rate token naturally aggregates information over a content-relevant temporal span even without external SSL or ASR supervisions. This is consistent with the restricted-reconstruction results in Table 3, where a 64-dimensional core can already support low WER, while its SIM remains much lower than full-dimensional token reconstruction. This suggests that the core representation is relatively content-centric and less complete acoustically. From this perspective, native low frame rate itself may act as a semantic inductive bias, reducing the need for externally imposed semantic alignment losses.

Third, the short-form and long-form results differ substantially. The effective generated durations on Seed-TTS-eval are only 5.28 ± 0.96 s for ZH and 4.18 ± 1.16 s for EN, whereas the long-form set has an effective duration of 52.14 ± 3.04 s. The Seed-TTS-eval results therefore mainly measure short-form quality and do not fully expose AR error accumulation. As shown in Figure 7, except for the two examples that exhibit severe error accumulation almost from the beginning, the remaining failure cases can still maintain stable synthesis for at least the first 6 seconds, corresponding to roughly 50 tokens, before visible drift emerges. This delayed failure mode is also reflected in Table 4, where many token and CFG configurations obtain similar WER on Seed-TTS-eval, but their long-form behavior differs sharply. Moreover, there also appears to be a dataset effect in the absolute SIM values: the first-segment SIM on the long-form set is typically about 0.05 higher than the ZH SIM on Seed-TTS-eval, and the best SIM observed on the medium-length CFG-selection set in Figure 6 is nearly 0.07 higher. This may partly reflect a domain mismatch rather than only a duration effect. Since our training data are dominated by real-world recordings, whereas Seed-TTS-eval is constructed from

curated benchmark datasets, differences in recording conditions, speaker characteristics, content style, and reference-audio distribution may therefore affect SIM evaluation. This provides one possible explanation for why the model appears stronger on internal real-world evaluation sets while showing a larger SIM gap on the standard benchmark.

Fourth, a moderate core dimension gives the best overall balance. Very small core dimensions provide stronger constraints and can help WER, but may reduce acoustic capacity; larger core dimensions preserve more information but are less constrained. Across training loss, short-form generation, and long-form stability, 32/✓ is the most balanced configuration: it achieves the best ZH WER on Seed-TTS-eval, 0.95%, and strong long-form performance under CFG-L, with WER 4.61%, gSIM 0.731, and a stable segment-SIM curve ending at 0.672. This agrees with the prediction-loss curves in Figure 5, where 32/✓ also reaches the lowest final direction loss.

Finally, PDD consistently improves the generation behavior, especially for long-form stability and SIM. Its effect is most visible under the more aggressive CFG-S setting. For example, changing 32/× to 32/✓ reduces long-form WER from 20.80% to 9.73%, increases gSIM from 0.674 to 0.724, and improves the fifth-segment SIM from 0.419 to 0.578. Similarly, changing 16/× to 16/✓ under CFG-S improves the fifth-segment SIM from 0.273 to 0.601. These results indicate that the PDD-induced coordinate hierarchy improves not only the optimization loss but also the robustness of AR rollout, supporting the proposed identifiability mechanism.

Table 5 further compares MP-ELD with prior AR TTS systems on Seed-TTS-eval. The proposed system is highly competitive in WER, while its SIM scores remain lower than those of the strongest prior systems. One possible explanation is that the token frame rate acts not only as an efficiency parameter, but also as a temporal-resolution knob, loosely analogous to the analysis-window length in time–frequency analysis. A lower token frame rate means that each token summarizes a longer span of audio, similar to how a larger FFT window provides a longer analysis context and higher frequency resolution, but becomes less sensitive to rapid temporal variation. Such longer-span aggregation can make phonetic or content-level structure more stable and can shorten the AR horizon, which may help WER and long-horizon stability. At the same time, it may reduce sensitivity to fine-grained local acoustic variations, such as micro-prosody, transient spectral details, and short-time speaker-specific cues, which are important for speaker similarity. Conversely, higher-rate tokens, or product-structured local representations that preserve multiple short-range sub-tokens within each LM step, can retain more local acoustic detail and therefore often achieve stronger SIM, but they usually require longer native sequences or additional local prediction modules.

However, VibeVoice suggests that this temporal-resolution interpretation might be incomplete, as it uses an even lower native token frame rate of 7.5 Hz but achieves substantially stronger SIM than MP-ELD. We hypothesize that this difference may come from its explicit semantic–acoustic tokenizer design. In such a factorized representation, the low-rate semantic component can still provide the long-span content aggregation and AR-stability benefits associated with low frame rate, while a more independent acoustic component can preserve speaker-specific and fine-grained acoustic information. By contrast, Locodec uses a single reconstruction-first continuous token space, so content, speaker identity, and local acoustic detail must be organized within the same low-rate high-dimensional token. Under AR prediction, this unified space may naturally favor content-stable components, helping WER, while making fine-grained acoustic similarity harder to model. This suggests a potential direction for future tokenizer design: introducing semantic–acoustic factorization into a low-rate continuous token space, possibly still without relying on explicit SSL or ASR supervision, may preserve the content clustering and stability benefits of native low-frame-rate tokens while improving acoustic fidelity and speaker similarity.

6 Conclusion and Future Work

In this paper, we studied whether low-frame-rate, high-dimensional continuous tokens can serve as stable targets for autoregressive speech generation. Our main finding is that such tokens are viable when the representation space and the generative framework are designed jointly. We proposed Locodec, a locally encoded tokenizer which shapes a spherical high-dimensional token space around a lower-dimensional core manifold and induces a coordinate-wise energy hierarchy through PDD, improving predictability without

noticeably degrading full-dimensional token reconstruction quality. We further proposed MP-ELD, which separates local-continuity, self-consistency, and alignment-consistency information pathways, allowing residual CFG to control acoustic consistency and content alignment more explicitly. The experiments show that reconstruction quality alone is not sufficient to characterize a generative representation, and token-space geometry strongly affects generator training loss, CFG behavior, and long-form stability. A moderate core dimension combined with PDD provides the best overall balance, and the resulting 8-Hz, 768-dimensional continuous tokens support competitive WER without external SSL/ASR models, pretrained text LMs, or post-training stages. At the same time, the remaining SIM gap to the strongest prior systems suggests a semantic–acoustic trade-off: low frame rate appears to favor content aggregation and AR stability, but may make fine-grained acoustic and speaker-similarity modeling harder. We therefore view token frame rate not only as an efficiency parameter, but also as a semantic–acoustic resolution knob.

Several extensions are natural. First, although Locodec is formulated as a general stereo-audio tokenizer framework and can be instantiated for higher sampling rates such as 48 kHz, the experiments in this paper focus primarily on 24-kHz speech reconstruction and the TTS task. We plan to further evaluate the Locodec framework on broader audio understanding and generation tasks and even more general multimodal settings. The same principles may also be relevant beyond audio: low-rate high-dimensional continuous tokens with shaped latent geometry could be useful for image, video, or other sequence generation problems where reconstruction capacity and AR stability must be balanced.

Second, our deliberate avoidance of pretrained SSL, ASR, and language models is a design choice for isolating the effects of token-space shaping and information routing, not a restriction of the method. Locodec and MP-ELD are naturally compatible with pretrained components. For example, the alignment-consistency LM could be initialized from or replaced by a pretrained text LM, which may strengthen the semantic bias of the alignment path and thereby alter the role of the self-consistency path. Similarly, the token encoders for different MP-ELD pathways could be initialized differently, or one pathway could be guided by a pretrained SSL or ASR representation. More generally, information routing can emerge from training dynamics, as shown in this paper, but it can also be strengthened by architectural and initialization priors. Designing better information pathways, both inside the tokenizer and inside the generator, can potentially be an important direction to further mitigate or even eliminate AR error accumulation.

Third, improving speaker similarity and acoustic fidelity remains a central challenge. The current results suggest that low-frame-rate tokens naturally favor content aggregation, which helps WER but may make fine-grained acoustic modeling harder. One possible direction is to introduce semantic–acoustic factorization into the token space, still without relying on explicit SSL or ASR supervision. More broadly, for general audio generation, the relevant factors may go beyond semantics and acoustics to include spatial attributes, source identity, or temporal event dynamics. Developing tokenizers that can organize these factors in a controllable way, while preserving reconstruction fidelity and AR predictability, is a key open problem.

Fourth, scaling the individual components of the proposed framework is a natural next step. In particular, since the decoder is the actual next-token prediction module that estimates the pathwise velocity field, understanding its scaling behavior is particularly important. Compared with product-structured token approaches, where each LM step may require an additional local DiT to decode a group of short-range tokens, the FFN decoder in MP-ELD requires significantly fewer MACs to predict one native high-dimensional token. This leaves substantial computational headroom for scaling the decoder or replacing it with more expressive architectures while still keeping the overall inference cost competitive.

Finally, the long-form experiments in this paper are still limited relative to truly long streaming scenarios. Minute-scale generation already reveals clear AR error accumulation, but hour-scale streaming, multi-speaker interaction, podcast-level synthesis, and complex acoustic scenes may introduce additional failure modes. Future work should therefore study MP-ELD under longer and more diverse generation regimes. In addition, the experiments show that the optimal CFG configuration depends on utterance length, token representation, and the desired trade-off between WER, SIM, and stability. This suggests that CFG may not necessarily be time-invariant during AR generation. Dynamic guidance schedules that vary not only with bridge time τ, but also with AR generation time, may provide a more flexible way to maintain both short-horizon quality and long-horizon stability.

References

[1] Michael Albergo, Nicholas M Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. Journal of Machine Learning Research, 26(209):1–80, 2025.

[2] Michael S Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. arXiv preprint arXiv:2209.15571, 2022.

[3] Naomi Altman and Martin Krzywinski. The curse (s) of dimensionality. Nat Methods, 15(6):399–400, 2018.

[4] Philip Anastassiou, Jiawei Chen, Jitong Chen, Yuanzhe Chen, Zhuo Chen, Ziyi Chen, Jian Cong, Lelai Deng, Chuang Ding, Lu Gao, et al. Seed-TTS: A family of high-quality versatile speech generation models. arXiv preprint arXiv:2406.02430, 2024.

[5] Rosana Ardila, Megan Branson, Kelly Davis, Michael Kohler, Josh Meyer, Michael Henretty, Reuben Morais, Lindsay Saunders, Francis Tyers, and Gregor Weber. Common voice: A massively-multilingual speech corpus. In Proceedings of the twelfth language resources and evaluation conference, pages 4218–4222, 2020.

[6] Georgios Arvanitidis, Lars Kai Hansen, and Søren Hauberg. Latent space oddity: on the curvature of deep generative models. arXiv preprint arXiv:1710.11379, 2017.

[7] Georgios Arvanitidis, Søren Hauberg, and Bernhard Schölkopf. Geometrically enriched latent spaces. arXiv preprint arXiv:2008.00565, 2020.

[8] Bishnu S Atal and Manfred R Schroeder. Adaptive predictive coding of speech signals. Bell System Technical Journal, 49(8):1973–1986, 1970.

[9] Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations. Advances in neural information processing systems, 33: 12449–12460, 2020.

[10] Ye Bai, Haonan Chen, Jitong Chen, Zhuo Chen, Yi Deng, Xiaohong Dong, Lamtharn Hantrakul, Weituo Hao, Qingqing Huang, Zhongyi Huang, et al. Seed-music: A unified framework for high quality and controlled music generation. arXiv preprint arXiv:2409.09214, 2024.

[11] Peter GJ Barten. Contrast sensitivity of the human eye and its effects on image quality. SPIE press, 1999.

[12] Eric Battenberg, RJ Skerry-Ryan, Soroosh Mariooryad, Daisy Stanton, David Kao, Matt Shannon, and Tom Bagby. Location-relative attention mechanisms for robust long-form speech synthesis. In ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 6194–6198. IEEE, 2020.

[13] Bruno Bessette, Redwan Salami, Roch Lefebvre, Milan Jelinek, Jani Rotola-Pukkila, Janne Vainio, Hannu Mikkola, and Kari Jarvinen. The adaptive multirate wideband speech codec (AMR-WB). IEEE transactions on speech and audio processing, 10(8):620–636, 2002.

[14] Zalán Borsos, Raphaël Marinier, Damien Vincent, Eugene Kharitonov, Olivier Pietquin, Matt Sharifi, Dominik Roblek, Olivier Teboul, David Grangier, Marco Tagliasacchi, et al. Audiolm: a language modeling approach to audio generation. IEEE/ACM transactions on audio, speech, and language processing, 31:2523–2533, 2023.

[15] Fergus W Campbell and John G Robson. Application of fourier analysis to the visibility of gratings. The Journal of physiology, 197(3):551, 1968.

[16] Olivier Chapelle, Bernhard Schölkopf, and Alexander Zien. A discussion of semi-supervised learning and transduction. In Semi-supervised learning, pages 473–478. MIT Press, 2006.

[17] Boyuan Chen, Diego Martí Monsó, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion forcing: Next-token prediction meets full-sequence diffusion. Advances in Neural Information Processing Systems, 37:24081–24125, 2024.

[18] Qian Chen, Yafeng Chen, Yanni Chen, Mengzhe Chen, Yingda Chen, Chong Deng, Zhihao Du, Ruize Gao, Changfeng Gao, Zhifu Gao, et al. Minmo: A multimodal large language model for seamless voice interaction. arXiv preprint arXiv:2501.06282, 2025.

[19] Ricky TQ Chen and Yaron Lipman. Flow matching on general geometries. arXiv preprint arXiv:2302.03660, 2023.

[20] Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, et al. Wavlm: Large-scale self-supervised pre-training for full stack speech processing. IEEE Journal of Selected Topics in Signal Processing, 16(6):1505–1518, 2022.

[21] Sanyuan Chen, Shujie Liu, Long Zhou, Yanqing Liu, Xu Tan, Jinyu Li, Sheng Zhao, Yao Qian, and Furu Wei. Vall-e 2: Neural codec language models are human parity zero-shot text to speech synthesizers. arXiv preprint arXiv:2406.05370, 2024.

[22] Ting Chen. On the importance of noise scheduling for diffusion models. arXiv preprint arXiv:2301.10972, 2023.

[23] Wenxi Chen, Dongya Jia, Yushen Chen, Zhikang Niu, Yuzhe Liang, Xiquan Li, Ruiqi Yan, Ziyang Ma, Guanrou Yang, Sanyuan Chen, et al. Wavtts: Towards high-quality zero-shot tts via direct raw waveform modeling. arXiv preprint arXiv:2606.03455, 2026.

[24] Wenxi Chen, Ruiqi Yan, Yushen Chen, Zhikang Niu, Ziyang Ma, Xiquan Li, Yuzhe Liang, Shunshun Yin, Ming Tao, Xinsheng Wang, et al. Sac: Neural speech codec with semantic-acoustic dual-stream quantization. In Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 3030–3048, 2026.

[25] Yushen Chen, Zhikang Niu, Ziyang Ma, Keqi Deng, Chunhui Wang, JianZhao JianZhao, Kai Yu, and Xie Chen. F5-tts: A fairytaler that fakes fluent and faithful speech with flow matching. In Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 6255–6271, 2025.

[26] Michael Chinen, Felicia SC Lim, Jan Skoglund, Nikita Gureev, Feargus O'Gorman, and Andrew Hines. Visqol v3: An open source production ready objective speech and audio metric. In 2020 twelfth international conference on quality of multimedia experience (QoMEX), pages 1–6. IEEE, 2020.

[27] Chung-Cheng Chiu, James Qin, Yu Zhang, Jiahui Yu, and Yonghui Wu. Self-supervised learning with random-projection quantizer for speech recognition. In International Conference on Machine Learning, pages 3915–3924. PMLR, 2022.

[28] Yunfei Chu, Jin Xu, Xiaohuan Zhou, Qian Yang, Shiliang Zhang, Zhijie Yan, Chang Zhou, and Jingren Zhou. Qwen-audio: Advancing universal audio understanding via unified large-scale audio-language models. arXiv preprint arXiv:2311.07919, 2023.

[29] Jade Copet, Felix Kreuk, Itai Gat, Tal Remez, David Kant, Gabriel Synnaeve, Yossi Adi, and Alexandre Défossez. Simple and controllable music generation. Advances in neural information processing systems, 36:47704–47720, 2023.

[30] Gregory A Daly, Jonathan E Fieldsend, and Gavin Tabor. Variational autoencoders without the variation. arXiv preprint arXiv:2203.00645, 2022.

[31] Tim R Davidson, Luca Falorsi, Nicola De Cao, Thomas Kipf, and Jakub M Tomczak. Hyperspherical variational auto-encoders. arXiv preprint arXiv:1804.00891, 2018.

[32] Friso de Kruiff, Erik Bekkers, Ozan Öktem, Carola-Bibiane Schönlieb, and Willem Diepeveen. Pullback flow matching on data manifolds. arXiv preprint arXiv:2410.04543, 2024.

[33] Alexandre Défossez, Jade Copet, Gabriel Synnaeve, and Yossi Adi. High fidelity neural audio compression. arXiv preprint arXiv:2210.13438, 2022.

[34] Alexandre Défossez, Laurent Mazaré, Manu Orsini, Amélie Royer, Patrick Pérez, Hervé Jégou, Edouard Grave, and Neil Zeghidour. Moshi: a speech-text foundation model for real-time dialogue. arXiv preprint arXiv:2410.00037, 2024.

[35] Prafulla Dhariwal, Heewoo Jun, Christine Payne, Jong Wook Kim, Alec Radford, and Ilya Sutskever. Jukebox: A generative model for music. arXiv preprint arXiv:2005.00341, 2020.

[36] Ding Ding, Zeqian Ju, Yichong Leng, Songxiang Liu, Tong Liu, Zeyu Shang, Kai Shen, Wei Song, Xu Tan, Heyi Tang, et al. Kimi-audio technical report. arXiv preprint arXiv:2504.18425, 2025.

[37] Zhihao Du, Qian Chen, Shiliang Zhang, Kai Hu, Heng Lu, Yexin Yang, Hangrui Hu, Siqi Zheng, Yue Gu, Ziyang Ma, et al. Cosyvoice: A scalable multilingual zero-shot text-to-speech synthesizer based on supervised semantic tokens. arXiv preprint arXiv:2407.05407, 2024.

[38] Zhihao Du, Changfeng Gao, Yuxuan Wang, Fan Yu, Tianyu Zhao, Hao Wang, Xiang Lv, Hui Wang, Chongjia Ni, Xian Shi, et al. Cosyvoice 3: Towards in-the-wild speech generation via scaling-up and post-training. arXiv preprint arXiv:2505.17589, 2025.

[39] Sefik Emre Eskimez, Xiaofei Wang, Manthan Thakker, Canrun Li, Chung-Hsien Tsai, Zhen Xiao, Hemin Yang, Zirun Zhu, Min Tang, Xu Tan, et al. E2 tts: Embarrassingly easy fully non-autoregressive zero-shot tts. In 2024 IEEE spoken language technology workshop (SLT), pages 682–689. IEEE, 2024.

[40] Zach Evans, Julian D Parker, CJ Carr, Zack Zukowski, Josiah Taylor, and Jordi Pons. Long-form music generation with latent diffusion. arXiv preprint arXiv:2404.10301, 2024.

[41] Wei Fan, Chao-Hong Tan, Qian Chen, Wen Wang, Xiangang Li, Kejiang Chen, Weiming Zhang, and Nenghai Yu. Barewave: Waveform-native flow-matching text-to-speech. arXiv preprint arXiv:2606.09048, 2026.

[42] Sreyan Ghosh, Arushi Goel, Jaehyeon Kim, Sonal Kumar, Zhifeng Kong, Sang-gil Lee, Chao-Han Yang, Ramani Duraiswami, Dinesh Manocha, Rafael Valle, et al. Audio flamingo 3: Advancing audio intelligence with fully open large audio language models. Advances in Neural Information Processing Systems, 38:41819–41886, 2026.

[43] Alexander N Gorban and Ivan Yu Tyukin. Blessing of dimensionality: mathematical foundations of the statistical physics of data. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 376(2118):20170237, 2018.

[44] Hao-Han Guo, Yao Hu, Kun Liu, Fei-Yu Shen, Xu Tang, Yi-Chen Wu, Feng-Long Xie, Kun Xie, and Kai-Tuo Xu. Fireredtts: A foundation text-to-speech framework for industry-level generative speech applications. arXiv preprint arXiv:2409.03283, 2024.

[45] Tingwei Guo, Cheng Wen, Dongwei Jiang, Ne Luo, Ruixiong Zhang, Shuaijiang Zhao, Wubo Li, Cheng Gong, Wei Zou, Kun Han, et al. Didispeech: A large scale mandarin speech corpus. In ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 6968–6972. IEEE, 2021.

[46] Cong Han, Yi Luo, Chenda Li, Tianyan Zhou, Keisuke Kinoshita, Shinji Watanabe, Marc Delcroix, Hakan Erdogan, John R Hershey, Nima Mesgarani, et al. Continuous speech separation using speaker inventory for long recording. In Interspeech, pages 3036–3040, 2021.

[47] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-VAE: Learning basic visual concepts with a constrained variational framework. In International Conference on Learning Representations, 2017. URL https://openreview.net/forum?id=Sy2fzU9gl.

[48] Emiel Hoogeboom, Thomas Mensink, Jonathan Heek, Kay Lamerigts, Ruiqi Gao, and Tim Salimans. Simpler diffusion: 1.5 fid on imagenet512 with pixel-space diffusion. In Proceedings of the Computer Vision and Pattern Recognition Conference, pages 18062–18071, 2025.

[49] Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. IEEE/ACM transactions on audio, speech, and language processing, 29:3451–3460, 2021.

[50] Cheng-Zhi Anna Huang, Ashish Vaswani, Jakob Uszkoreit, Noam Shazeer, Ian Simon, Curtis Hawthorne, Andrew M Dai, Matthew D Hoffman, Monica Dinculescu, and Douglas Eck. Music transformer. arXiv preprint arXiv:1809.04281, 2018.

[51] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. arXiv preprint arXiv:2410.21276, 2024.

[52] Shengpeng Ji, Ziyue Jiang, Wen Wang, Yifu Chen, Minghui Fang, Jialong Zuo, Qian Yang, Xize Cheng, Ruiqi Li, Ziang Zhang, et al. Wavtokenizer: an efficient acoustic discrete codec tokenizer for audio language modeling. In International Conference on Learning Representations, volume 2025, pages 93809–93826, 2025.

[53] Dongya Jia, Zhuo Chen, Jiawei Chen, Chenpeng Du, Jian Wu, Jian Cong, Xiaobin Zhuang, Chumin Li, Zhen Wei, Yuping Wang, et al. Ditar: Diffusion transformer autoregressive modeling for speech generation. arXiv preprint arXiv:2502.03930, 2025.

[54] Yuepeng Jiang, Huakang Chen, Ziqian Ning, Jixun Yao, Zerui Han, Di Wu, Meng Meng, Jian Luan, Zhonghua Fu, and Lei Xie. Diffrhythm 2: Efficient and high fidelity song generation via block flow matching. arXiv preprint arXiv:2510.22950, 2025.

[55] Zeqian Ju, Yuancheng Wang, Kai Shen, Xu Tan, Detai Xin, Dongchao Yang, Yanqing Liu, Yichong Leng, Kaitao Song, Siliang Tang, et al. Naturalspeech 3: Zero-shot speech synthesis with factorized codec and diffusion models. arXiv preprint arXiv:2403.03100, 2024.

[56] Guolin Ke and Hui Xue. Hyperspherical latents improve continuous-token autoregressive generation. arXiv preprint arXiv:2509.24335, 2025.

[57] Robert Kubichek. Mel-cepstral distance measure for objective speech quality assessment. In Proceedings of IEEE pacific rim conference on communications computers and signal processing, volume 1, pages 125–128. IEEE, 1993.

[58] Rithesh Kumar, Prem Seetharaman, Alejandro Luebs, Ishaan Kumar, and Kundan Kumar. High-fidelity audio compression with improved RVQGAN. Advances in Neural Information Processing Systems, 36:27980–27993, 2023.

[59] Doyup Lee, Chiheon Kim, Saehoon Kim, Minsu Cho, and Wook-Shin Han. Autoregressive image generation using residual quantization. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 11523–11532, 2022.

[60] Junho Lee, Kwanseok Kim, and Joonseok Lee. Geometry-aware image flow matching. arXiv preprint arXiv:2605.25294, 2026.

[61] Sangyun Lee, Gayoung Lee, Hyunsu Kim, Junho Kim, and Youngjung Uh. Sequential data generation with groupwise diffusion process. arXiv preprint arXiv:2310.01400, 2023.

[62] Xingjian Leng, Jaskirat Singh, Yunzhong Hou, Zhenchang Xing, Saining Xie, and Liang Zheng. REPA-E: Unlocking VAE for end-to-end tuning of latent diffusion transformers. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 18262–18272, 2025.

[63] Tianhong Li and Kaiming He. Back to basics: Let denoising generative models denoise. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 36115–36125, 2026.

[64] Tianhong Li, Yonglong Tian, He Li, Mingyang Deng, and Kaiming He. Autoregressive image generation without vector quantization. Advances in Neural Information Processing Systems, 37:56424–56445, 2024.

[65] Shi Lian, Changtao Li, Bohan Li, Hankun Wang, Da Zheng, Junfeng Tian, Yufeng Ma, Colin Zhang, and Kai Yu. dots. tts technical report. arXiv preprint arXiv:2606.07080, 2026.

[66] Shijia Liao, Yuxuan Wang, Songting Liu, Yifan Cheng, Ruoyi Zhang, Tianyu Li, Shidong Li, Yisheng Zheng, Xingwei Liu, Qingzheng Wang, et al. Fish audio s2 technical report. arXiv preprint arXiv:2603.08823, 2026.

[67] Haohe Liu, Xuenan Xu, Yi Yuan, Mengyue Wu, Wenwu Wang, and Mark D Plumbley. Semanticodec: An ultra low bitrate semantic audio codec for general sound. IEEE Journal of Selected Topics in Signal Processing, 18 (8):1448–1461, 2024.

[68] Zhijun Liu, Shuai Wang, Sho Inoue, Qibing Bai, and Haizhou Li. Autoregressive diffusion transformer for text-to-speech synthesis. arXiv preprint arXiv:2406.05551, 2024.

[69] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 11976–11986, 2022.

[70] Yi Luo, Zhuo Chen, and Takuya Yoshioka. Dual-path RNN: Efficient long sequence modeling for time-domain single-channel speech separation. In ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 46–50. IEEE, 2020.

[71] Yi Luo, Cong Han, and Nima Mesgarani. Group communication with context codec for lightweight source separation. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 29:1752–1761, 2021.

[72] Yi Luo, Jianwei Yu, Hangting Chen, Rongzhi Gu, and Chao Weng. Gull: A generative multifunctional audio codec. arXiv preprint arXiv:2404.04947, 2024.

[73] Huanru Henry Mao, Shuyang Li, Julian McAuley, and Garrison Cottrell. Speech recognition and multi-speaker diarization of long conversations. arXiv preprint arXiv:2005.08072, 2020.

[74] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In Proceedings of the IEEE international conference on computer vision, pages 2794–2802, 2017.

[75] Soroush Mehri, Kundan Kumar, Ishaan Gulrajani, Rithesh Kumar, Shubham Jain, Jose Sotelo, Aaron Courville, and Yoshua Bengio. SampleRNN: An unconditional end-to-end neural audio generation model. arXiv preprint arXiv:1612.07837, 2016.

[76] Lingwei Meng, Long Zhou, Shujie Liu, Sanyuan Chen, Bing Han, Shujie Hu, Yanqing Liu, Jinyu Li, Sheng Zhao, Xixin Wu, et al. Autoregressive speech synthesis without vector quantization. In Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1287–1300, 2025.

[77] Fabian Mentzer, David Minnen, Eirikur Agustsson, and Michael Tschannen. Finite scalar quantization: VQ-VAE made simple. In International Conference on Learning Representations, volume 2024, pages 51772–51783, 2024.

[78] Tuna Han Salih Meral, Kaan Oktay, Hidir Yesiltepe, Adil Kaan Akan, and Pinar Yanardag. Aligning latent geometry for spherical flow matching in image generation. arXiv preprint arXiv:2605.15193, 2026.

[79] George A Miller. Sensitivity to changes in the intensity of white noise and its relation to masking and loudness. The Journal of the Acoustical Society of America, 19(4):609–619, 1947.

[80] Pooneh Mousavi, Jarod Duret, Salah Zaiem, Luca Della Libera, Artem Ploujnikov, Cem Subakan, and Mirco Ravanelli. How should we extract discrete audio tokens from self-supervised models? arXiv preprint arXiv:2406.10735, 2024.

[81] Arun Narayanan, Rohit Prabhavalkar, Chung-Cheng Chiu, David Rybach, Tara N Sainath, and Trevor Strohman. Recognizing long-form speech using streaming end-to-end models. In 2019 IEEE automatic speech recognition and understanding workshop (ASRU), pages 920–927. IEEE, 2019.

[82] Mang Ning, Enver Sangineto, Angelo Porrello, Simone Calderara, and Rita Cucchiara. Input perturbation reduces exposure bias in diffusion models. arXiv preprint arXiv:2301.11706, 2023.

[83] Ziqian Ning, Huakang Chen, Yuepeng Jiang, Chunbo Hao, Guobin Ma, Shuai Wang, Jixun Yao, and Lei Xie. Diffrhythm: Blazingly fast and embarrassingly simple end-to-end full-length song generation with latent diffusion. arXiv preprint arXiv:2503.01183, 2025.

[84] Julian Parker, Anton Smirnov, Jordi Pons, CJ Carr, Zack Zukowski, Zach Evans, and Xubo Liu. Scaling transformers for low-bitrate high-quality speech coding. In International Conference on Learning Representations, volume 2025, pages 51997–52021, 2025.

[85] William Peebles and Saining Xie. Scalable diffusion models with transformers. In Proceedings of the IEEE/CVF international conference on computer vision, pages 4195–4205, 2023.

[86] Zhiliang Peng, Jianwei Yu, Wenhui Wang, Yaoyao Chang, Yutao Sun, Li Dong, Yi Zhu, Weijiang Xu, Hangbo Bao, Zehua Wang, et al. Vibevoice technical report. arXiv preprint arXiv:2508.19205, 2025.

[87] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. In International conference on machine learning, pages 28492–28518. PMLR, 2023.

[88] Desh Raj, Pavel Denisov, Zhuo Chen, Hakan Erdogan, Zili Huang, Maokui He, Shinji Watanabe, Jun Du, Takuya Yoshioka, Yi Luo, et al. Integration of speech separation, diarization, and recognition for multi-speaker meetings: System description, comparison, and analysis. In 2021 IEEE spoken language technology workshop (SLT), pages 897–904. IEEE, 2021.

[89] David Ruhe, Jonathan Heek, Tim Salimans, and Emiel Hoogeboom. Rolling diffusion models. arXiv preprint arXiv:2402.09470, 2024.

[90] Florian Schmidt. Generalization in generation: A closer look at exposure bias. In Proceedings of the 3rd Workshop on Neural Generation and Translation, pages 157–167, 2019.

[91] Manfred Schroeder and B Atal. Code-excited linear prediction (CELP): High-quality speech at very low bit rates. In ICASSP'85. IEEE International Conference on Acoustics, Speech, and Signal Processing, volume 10, pages 937–940. IEEE, 1985.

[92] Noam Shazeer. Glu variants improve transformer. arXiv preprint arXiv:2002.05202, 2020.

[93] Yangyang Shi, Yongqiang Wang, Chunyang Wu, Ching-Feng Yeh, Julian Chan, Frank Zhang, Duc Le, and Mike Seltzer. Emformer: Efficient memory transformer based acoustic model for low latency streaming speech recognition. In ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 6783–6787. IEEE, 2021.

[94] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. arXiv preprint arXiv:2011.13456, 2020.

[95] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. Neurocomputing, 568:127063, 2024.

[96] Xingzhi Sun, Danqi Liao, Kincaid MacDonald, Yanlei Zhang, Guillaume Huguet, Guy Wolf, Ian Adelstein, Tim GJ Rudner, and Smita Krishnaswamy. Geometry-aware autoencoders for metric learning and generative modeling on data manifolds. In ICML 2024 Workshop on Geometry-grounded Representation Learning and Generative Modeling, 2024.

[97] Cees H Taal, Richard C Hendriks, Richard Heusdens, and Jesper Jensen. An algorithm for intelligibility prediction of time–frequency weighted noisy speech. IEEE Transactions on audio, speech, and language processing, 19(7): 2125–2136, 2011.

[98] Gemma Team. Gemma 4 technical report, 2026. URL https://arxiv.org/abs/2607.02770.

[99] Shengbang Tong, Boyang Zheng, Ziteng Wang, Bingda Tang, Nanye Ma, Ellis Brown, Jihan Yang, Rob Fergus, Yann LeCun, and Saining Xie. Scaling text-to-image diffusion transformers with representation autoencoders. arXiv preprint arXiv:2601.16208, 2026.

[100] Aaron Van Den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, Koray Kavukcuoglu, et al. Wavenet: A generative model for raw audio. arXiv preprint arXiv:1609.03499, 12(1), 2016.

[101] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. Advances in neural information processing systems, 30, 2017.

[102] Chengyi Wang, Sanyuan Chen, Yu Wu, Ziqiang Zhang, Long Zhou, Shujie Liu, Zhuo Chen, Yanqing Liu, Huaming Wang, Jinyu Li, et al. Neural codec language models are zero-shot text to speech synthesizers. arXiv preprint arXiv:2301.02111, 2023.

[103] Shuai Wang, Ziteng Gao, Chenhui Zhu, Weilin Huang, and Limin Wang. Pixnerd: Pixel neural field diffusion. arXiv preprint arXiv:2507.23268, 2025.

[104] Xinsheng Wang, Mingqi Jiang, Ziyang Ma, Ziyu Zhang, Songxiang Liu, Linqin Li, Zheng Liang, Qixi Zheng, Rui Wang, Xiaoqin Feng, et al. Spark-tts: An efficient llm-based text-to-speech model with single-stream decoupled speech tokens. arXiv preprint arXiv:2503.01710, 2025.

[105] Yuancheng Wang, Zhenyu Tang, Yun Wang, Arthur Hinsvark, Yingru Liu, Yinghao Li, Kainan Peng, Junyi Ao, Mingbo Ma, Mike Seltzer, et al. Scaling speech tokenizers with diffusion autoencoders. arXiv preprint arXiv:2602.06602, 2026.

[106] Andrew B Watson and Joshua A Solomon. Model of visual contrast gain control and pattern masking. Journal of the optical society of America A, 14(9):2379–2391, 1997.

[107] David Wessels, David Knigge, Riccardo Valperga, Samuele Papa, Sharvaree Vadgama, Efstratios Gavves, and Erik Bekkers. Grounding continuous representations in geometry: Equivariant neural fields. In International Conference on Learning Representations, volume 2025, pages 59774–59794, 2025.

[108] Boyong Wu, Chao Yan, Chen Hu, Cheng Yi, Chengli Feng, Fei Tian, Feiyu Shen, Gang Yu, Haoyang Zhang, Jingbei Li, et al. Step-audio 2 technical report. arXiv preprint arXiv:2507.16632, 2025.

[109] Shengqiong Wu, Hao Fei, Leigang Qu, Wei Ji, and Tat-Seng Chua. Next-gpt: Any-to-any multimodal llm. arXiv preprint arXiv:2309.05519, 2023.

[110] Kun Xie, Feiyu Shen, Junjie Li, Fenglong Xie, Xu Tang, and Yao Hu. Fireredtts-2: Towards long conversational speech generation for podcast and chatbot. arXiv preprint arXiv:2509.02020, 2025.

[111] Detai Xin, Shujie Hu, Chengzuo Yang, Chen Huang, Guoqiao Yu, Guanglu Wan, and Xunliang Cai. Longcat-audiodit: High-fidelity diffusion text-to-speech in the waveform latent space. arXiv preprint arXiv:2603.29339, 2026.

[112] Jin Xu, Zhifang Guo, Hangrui Hu, Yunfei Chu, Xiong Wang, Jinzheng He, Yuxuan Wang, Xian Shi, Ting He, Xinfa Zhu, et al. Qwen3-omni technical report. arXiv preprint arXiv:2509.17765, 2025.

[113] Tongda Xu, Mingwei He, Shady Abu-Hussein, Jose Miguel Hernandez-Lobato, Chunhang Zheng, Kai Zhao, Chao Zhou, Ya-Qin Zhang, and Yan Wang. Making reconstruction FID predictive of diffusion generation FID. arXiv preprint arXiv:2603.05630, 2026.

[114] Chen Yang, Chufan Yu, Hanfu Chen, Jie Zhu, Jingqi Chen, Ke Chen, Wenxuan Wang, Yang Wang, Yaozhou Jiang, Yi Jiang, et al. Moss-audio technical report. arXiv preprint arXiv:2606.01802, 2026.

[115] Zhen Ye, Peiwen Sun, Jiahe Lei, Hongzhan Lin, Xu Tan, Zheqi Dai, Qiuqiang Kong, Jianyi Chen, Jiahao Pan, Qifeng Liu, et al. Codec does matter: Exploring the semantic shortcoming of codec for audio language model. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 39, pages 25697–25705, 2025.

[116] Tianwei Yin, Qiang Zhang, Richard Zhang, William T Freeman, Fredo Durand, Eli Shechtman, and Xun Huang. From slow bidirectional to fast autoregressive video diffusion models. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 22963–22974, 2025.

[117] Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. arXiv preprint arXiv:2410.06940, 2024.

[118] Yongsheng Yu, Wei Xiong, Weili Nie, Yichen Sheng, Shiqiu Liu, and Jiebo Luo. Pixeldit: Pixel diffusion transformers for image generation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 14273–14282, 2026.

[119] Ruibin Yuan, Hanfeng Lin, Shuyue Guo, Ge Zhang, Jiahao Pan, Yongyi Zang, Haohe Liu, Yiming Liang, Wenye Ma, Xingjian Du, et al. Yue: Scaling open foundation models for long-form music generation. arXiv preprint arXiv:2503.08638, 2025.

[120] Zhengrong Yue, Taihang Hu, Mengting Chen, Haiyu Zhang, Zihao Pan, Tao Liu, Zikang Wang, Jinsong Lan, Xiaoyong Zhu, Bo Zheng, et al. What matters for diffusion-friendly latent manifold? prior-aligned autoencoders for latent diffusion. arXiv preprint arXiv:2605.07915, 2026.

[121] Neil Zeghidour, Alejandro Luebs, Ahmed Omran, Jan Skoglund, and Marco Tagliasacchi. Soundstream: An end-to-end neural audio codec. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 30: 495–507, 2021.

[122] Dong Zhang, Gang Wang, Jinlong Xue, Kai Fang, Liang Zhao, Rui Ma, Shuhuai Ren, Shuo Liu, Tao Guo, Weiji Zhuang, et al. Mimo-audio: Audio language models are few-shot learners. arXiv preprint arXiv:2512.23808, 2025.

[123] Xin Zhang, Dong Zhang, Shimin Li, Yaqian Zhou, and Xipeng Qiu. Speechtokenizer: Unified speech tokenizer for speech language models. In International Conference on Learning Representations, volume 2024, pages 31798–31818, 2024.

[124] Boyang Zheng, Nanye Ma, Shengbang Tong, and Saining Xie. Diffusion transformers with representation autoencoders. arXiv preprint arXiv:2510.11690, 2025.

[125] Feiyan Zhou, Luyuan Wang, Shoufa Chen, Zhe Wang, Zhiheng Liu, Yuren Cong, Xiaohui Zhang, Fanny Yang, and Belinda Zeng. Wavflow: Audio generation in waveform space. arXiv preprint arXiv:2605.18749, 2026.

[126] Yixuan Zhou, Guoyang Zeng, Xin Liu, Xiang Li, Renjie Yu, Ziyang Wang, Runchuan Ye, Weiyue Sun, Jiancheng Gui, Kehan Li, et al. Voxcpm: Tokenizer-free tts for context-aware speech generation and true-to-life voice cloning. arXiv preprint arXiv:2509.24650, 2025.

[127] Yixuan Zhou, Guoyang Zeng, Xin Liu, Xiang Li, Renjie Yu, Jiancheng Gui, Jiaheng Wu, Ziyang Wang, Xudong Shen, Runchuan Ye, et al. Voxcpm2 technical report. arXiv preprint arXiv:2606.06928, 2026.

[128] Eberhard Zwicker and Hugo Fastl. Psychoacoustics: Facts and models, volume 22. Springer Science & Business Media, 2013.